User’s Guide

Publication number B3740-97005
Sixth edition, February 1997

For Safety information, Warranties and Regulatory information, see the pages following the Index.

HP B3740A Software Analyzer

Viewing Source Code for Logic Analysis Data

Using the Software Analyzer, you can control modules in the HP 16500 logic analysis system or the HP 1660C/CS and HP 1670A/D families of benchtop logic analyzers. You can also view the program source code associated with captured logic analyzer trace data.

The Software Analyzer works with state analyzers and preprocessor subsystems in HP logic analyzers on the Local Area Network (LAN). The HP 16500B Logic Analysis System mainframe must include a LAN interface card. The HP 1660C/CS families of logic analyzers must include the LAN interface (option 015).

With the Software Analyzer, you can:

- Use the Logic Analyzer window to set up traces and control the logic analyzer.
- Use the Source window to download symbolic information from OMF files to the logic analyzer.
- Use the Source window to view captured execution in terms of high-level "C" source lines. Other high-level programming languages may work depending on the symbolic debugging information in the OMF file.
- Use the Trace List Control window to scroll through the lines in the Source window as well as the state listing lines in the Logic Analyzer window.

In This Book

This book shows you how to use the Software Analyzer. It contains the following chapters:

- Chapter 1 is a brief tutorial that uses sample files to show you the steps you'll generally take when using the Software Analyzer.
- Chapter 2 contains an overview of the Software Analyzer, and has a more complete description of all the tasks you can perform when using the Software Analyzer.
- Chapter 3 describes the commands in the Software Analyzer main window, the HP E2450A Symbol Utility, supported symbol file formats, the nsbuild symbol utility, the General-Purpose ASCII (GPA) file format, and messages that can appear when using the Software Analyzer.
- Chapter 4 describes solutions for common problems that can occur when using the Software Analyzer, and why they happen.
- Chapter 5 shows you how to install the Software Analyzer. It contains instructions for personal computers running MS Windows® as well as for SUN and HP 9000 series 700 workstations running the UNIX® operating system.

Contents

1

This chapter uses sample configuration files and a demonstration program to configure the logic analyzer for the tutorial. For your convenience, the demonstration program is already compiled and linked.

This chapter includes the following steps to help you learn to use the Software Analyzer:

- Step 1. Start the Software Analyzer
- Step 2. Configure the logic analyzer
- Step 3. Load the symbol database
- Step 4. Set up the trigger and capture data
- Step 5. Scroll data by assembly states
- Step 6. Scroll data by source lines

Tutorial

Tutorial

Tutorial Works with Most HP Logic Analyzer Models

This tutorial will not work with the HP 1663C/CS 34-channel logic analyzer.

Before following the instructions in this tutorial, do the following.

- If you are using the HP 16500B logic analyzer mainframe, install the appropriate state analyzer module, with software version 2.00 or greater, into the mainframe. Refer to the documentation that comes with the logic analyzer module.
- If you are using the HP 16500B mainframe, install the HP 16500H or HP 16500L LAN Interface Module into the mainframe. Refer to the appropriate logic analyzer documentation.
- If you are using the HP 1660C/CS-Series Logic Analyzer, verify that it has LAN interface hardware (option 015) installed.
- Connect the logic analyzer to your LAN. Set the controller to Ethernet (in the System External I/O menu on benchtop models, and in the System Configuration menu for the HP 16500B/C).
- Install the appropriate logic analyzer system software. See the table on the following page and refer to the "Installation" chapter.

The E2450 Symbol Utility Software ships with the benchtop logic analyzer models and with the Software Analyzer.

- Install the Software Analyzer.
- Install the HP E2450A Symbol Utility Software in the logic analyzer.

See the "Installation" chapter for information about installing software.

Table 1-1 **HP Logic Analyzer Products and Software Version Numbers**

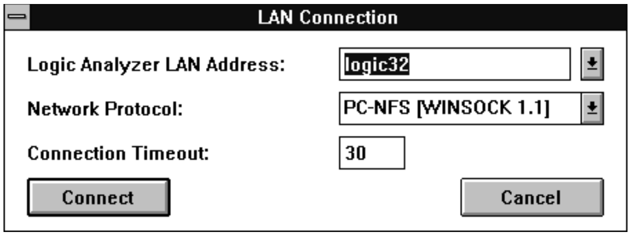
Product	Software Version
HP B3740A Software Analyzer	1.00 or greater
HP 16500C Mainframe	1.00 or greater
HP E2450A Symbol Utility Software for HP 16500C (OPT5_032)	1.00 or greater
HP 16500B Mainframe	3.00 or greater*
HP E2450A Symbol Utility Software for HP 16500B (OPT_032)	3.00 or greater*
HP 1660A/AS/C/CS-Series Logic Analyzer	1.00 or greater *
HP 1670A/D-Series Logic Analyzer	1.00 or greater *
HP E2450A Symbol Utility Software for HP 1660C/CS and 1670 (OPTC_032)	1.00 or greater*

* The Symbol Utility Software major software version number that you load must match tyour HP logic analyzer major software version number (for example, 3.xx) for the 16500B and 1.xx for the 16500C and 1660/70 series. The HP E2450A disk (labeled 3.xx) has both 1.xx and 3.xx files, for all the supported logic analyzers listed above.

Step 1. Start the Software Analyzer

1 Start the Software Analyzer by double-clicking on the Software Analyzer icon (in MS Windows®, Windows NT™, or Windows 95) or by typing "swa" on the command line (or "swa &" to start it as a background task in UNIX®).

2 In the "LAN Connection" dialog box, enter the logic analyzer LAN address (or select it from the list of previously entered names).


LAN Connection

Logic Analyzer LAN Address: logic32

Network Protocol: PC-NFS [WINSOCK 1.1]

Connection Timeout: 30

To find the LAN address, for an HP 16500B select the Communications field in the System Configuration menu then select the Ethernet configuration field. For an HP 16500C, select LAN Settings in the System Configuration menu. For an HP 1660C/CS-series or HP 1670-series logic analyzer, go to the System External I/O menu and select the LAN Settings field. The resulting menu shows the Internet (IP) Address of the logic analyzer.

3 **If you are using the Software Analyzer on a personal computer running MS Windows 3.1, select the appropriate "Network Protocol". This step is not required for personal computers running Windows NT or Windows 95.**

4 Click the **`Connect`** button.

Four Software Analyzer windows appear after a short time:

- The main Control window labeled "Software Analyzer" that contains three pulldown menus.
- The Logic Analyzer window labeled "HP Logic Analyzer-<analyzer name | IP address>".
- The Source code window labeled "Trace List Source".
- A window with trace list navigation buttons labeled "Trace List Control".

If you cannot connect to the logic analyzer, refer to "If you have LAN connection problems" in the Common Problems chapter.

Step 2. Configure the logic analyzer

Normally, after starting the Software Analyzer you should set up the analyzer configuration and format for a specific preprocessor. For this tutorial, the analyzer configuration is included with the product. All you need to do is load the configuration file.

1 **In the main Software Analyzer window, choose the File→Download... command.**

2 In the File Download dialog box, download the appropriate sample configuration file, with the full path, as shown in the following table. Enter the configuration file name in the Local File box, then click the `Download` button.

Table 1-2 **Sample Configuration Files and Uses**

Sample Configuration File	Use with Logic Analyzer/ Module	Enter this Command in the File Box
m332_550._a	HP 1660C/CS-Series, HP 16550A	C:\TMP\M332_550._A
m332_555._a	HP 16555, HP 16556	C:\TMP\M332_555._A
m332_554._a	HP 1670-Series, HP 16554A	C:\TMP\M332_554._A

If you installed the Software Analyzer in its default path, you can find the sample configuration files either in \HPCOL\SAMPLES\SWA on the personal computer, or in /usr/hpcol/<platform>/samples/swa on the UNIX workstation. You can click on the **`Browse...`** button and use the resulting dialog to select the Local File.

The sample files whose names contain "m332" or "332" are for the Motorola 68332 demonstration. Files whose names contain "i386" or "386" are Intel 80386 demonstration files. The "readme" file(s) in the sample files directories provide more information about the sample files and how they were generated.

3 In the File Download dialog box, type "icpu32" or "ia386e" in the Local File field. Type "C:\TMP\" in the Logic Analyzer File box. Click the `Download` button.

The Motorola demonstration uses "icpu32" and the Intel 80386 demonstration uses "ia386e".

4 Go to the System Hard Disk menu.

5 If the present logic analyzer directory is not \TMP, scroll through the list of files, highlight the appropriate directory name, and click on **`Change Dir`**.

To verify the present working directory, see the PWD: indicator line at the bottom of the Logic Analyzer window.

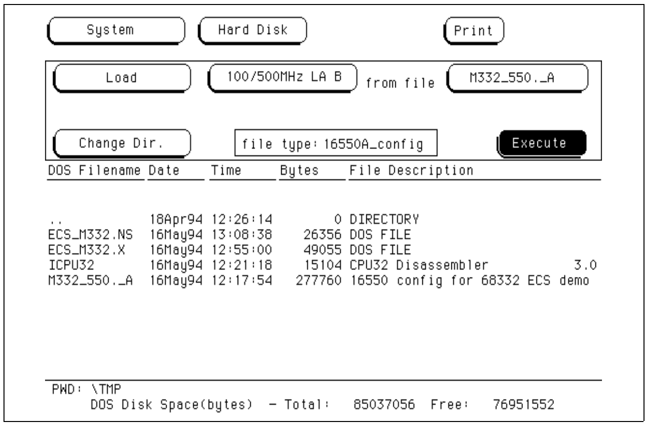
You can scroll through the list of files if you click on the graphical knob buttons and use the "shift-arrow keys", or if you hold down the right mouse button and drag the mouse. Use of the arrow keys may differ from use on a personal computer or in X-Windows.

6 **Scroll through the list of files in the \TMP directory and highlight the configuration file you just downloaded.**

7 **In the System Hard Disk screen (with `Load` showing), click on the module selection field (next to `Load`) and choose the appropriate module to load, or "analyzer" for benchtop models. Click on `Execute`. Note the following figure.**

Depending on the logic analyzer model you are using, when the load process finishes, the analyzer may display the following status messages: "Pods have been truncated. Clock and least significant pods have been preserved."

This figure shows an example of loading the M332_550._A configuration file into the HP 16550A State/Timing Logic Analyzer Module:



```
Load   100/500MHz LA B   from file   M332_550._A

file type: 16550A_config

DOS Filename Date    Time     Bytes  File Description

..           18Apr94 12:26:14      0 DIRECTORY
ECS_M332.NS  16May94 13:08:38  26356 DOS FILE
ECS_M332.X   16May94 12:55:00  49055 DOS FILE
ICPU32       16May94 12:21:18  15104 CPU32 Disassembler          3.0
M332_550._A  16May94 12:17:54 277760 16550 config for 68332 ECS demo

PWD: \TMP
     DOS Disk Space(bytes)  - Total:   85037056  Free:   76951552
```

The configuration file contains label setup, IAL setup, and previously stored trace data.

Step 3. Load the symbol database

An Environmental Control System (ECS) demonstration program is used for this tutorial. For your convenience, it is already compiled and linked into the Object Module Format (OMF) file named "ecs_m332.x". You must transfer this OMF file to the logic analyzer to use its symbols.

1 In the File Download dialog box entry fields, type the local file "\HPCOL\SAMPLES\SWA\ECS_M332.X" (for personal computers) or "/usr/hpcol/<platform>/samples/swa/ecs_m332.x" (for UNIX). Then, type the logic analyzer file "C:\TMP\ECS_M332.X".

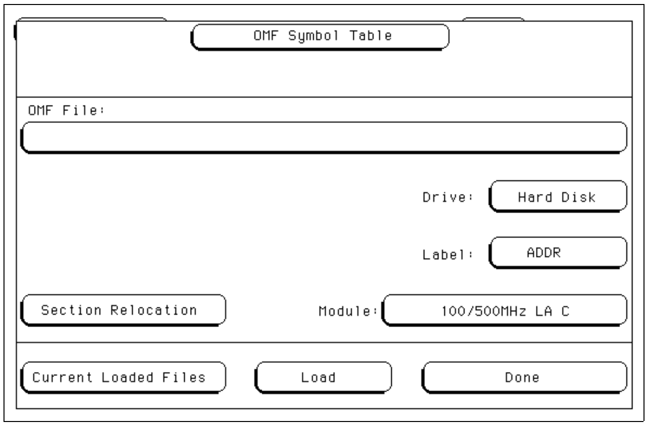
2 Click the **`Download`** button. When the download process finishes, click the **`Close`** button to close the File Download dialog box.

When the transfer completes, you can use the HP E2450A Symbol Utility to build a symbol database from this OMF file.

3 To enter the logic analyzer Symbol Utility subsystem, change from **`System`** to **`Symbol Utility`** in the Logic Analyzer window. If you are using an HP 1660C/CS-series or HP 1670-series logic analyzer, change from **`System`** to **`Symbols`**. Click on **`Specify Database`**.

The main OMF Symbol Table screen appears. You can perform multiple functions related to setting up the symbol database. The first thing to do is enter the name of the file that was just downloaded.

4 In OMF Symbol Table screen, click on the **`OMF File`** entry field.



OMF Symbol Table

OMF File:

Drive: Hard Disk

Label: ADDR

Module: 100/500MHz LA C

5 Load the "ECS_M332.X" file.

a Scroll to the \TMP directory and click on **`Select`**.

b Scroll through the list of files and highlight the "ECS_M332.X" file. Then, click on **`Select`**.

The next two entries rarely change. You can change the Drive entry to read files directly from the floppy drive. For this tutorial, keep it set to Hard Disk.

Always set the Label entry to "ADDR" for software traces. You can set this label to any arbitrary logic analyzer label. This is useful to allow hardware developers to read in hardware labels using the general purpose ASCII reader in the HP E2450A Symbol Utility. But, for normal software traces, you must always set this field to ADDR.

6 Click on the Module field and choose the appropriate module.

If there are multiple state analysis modules in an HP 16500 mainframe, the symbol manager must recognize the state analysis module to which this OMF symbol file attaches. This allows multiple programs to be traced at the same time using two different analysis modules and two different symbol files.

7 To build the symbol database, click on **`Load`**.

The logic analyzer builds an internal symbol database from the OMF file. An onscreen gauge informs you of the build progress.

After the build completes, you can view a trace list that has symbols in the display output.

You can also use the nsbuild utility to generate symbol databases. Refer to chapter 2 for information on the nsbuild utility.

8 In the OMF Symbol Table Screen, click on **`Done`**.

Step 4. Set up the trigger and capture data

Normally, you should next set up the trigger and run the analyzer to capture trace data. For this tutorial, trace data for the demonstration program was loaded when the analyzer configuration file loaded.

- To display the trace listing, click on Symbol Utility and choose the appropriate subsystem module (the module is always **`Analyzer`** in the benchtop logic analyzers). Then, click on **`Configuration`** and choose **`Listing`**.

A trace list, with symbols in the ADDR column, appears in the Logic Analyzer window. The address of the highlighted state corresponds to the function "main+000C" in the file "main_ecs.c" in the OMF file ECS_M332.X.

Also, the Trace List Source window becomes active. Its highlighted bar tracks the highlighted state in the trace list and shows the corresponding source code. In the Trace List Source window, you see the source code for file "main_ecs.c" (displayed in the title bar), and the highlighted bar appears in the middle of the function "main( )".

The trace listing may show a product-specific name when displayed on various logic analyzers.

Step 5. Scroll data by assembly states

To make it easy to view the trace data, the Software Analyzer has a "Trace List Control" window. This window contains four buttons that make it easier to scroll through the assembly and source views of the trace.

- To scroll to the next state in the trace listing, click the **`Next State`** button in Trace List Control window.

- To scroll to the previous state in the trace listing, click the **`Prev State`** button in Trace List Control window.

Notice that if you click the **`Next State`** button several times, the highlighted bar in the Trace List Source window also moves to a new corresponding source line.

Use the **`Next State`** and **`Prev State`** buttons when you wish to scroll one state at a time through the assembly level view.

Step 6. Scroll data by source lines

If you want to use the source code view, you can use the Trace List Control window's **`Next Line`** and **`Prev Line`** buttons to scroll through the state listing in source line increments.

- **To scroll to the state corresponding to the next source line, click the `Next Line` button in Trace List Control window.**

- **To scroll to the state corresponding to the previous source line, click the `Prev Line` button in Trace List Control window.**

When you scroll through source lines, the Software Analyzer searches the state listing until it finds the next state with a corresponding source line in the symbol database. When this is found, the state listing repositions to that state and the source window updates to display the matching source line.

You can also scroll backward in source line increments which can be very useful in understanding the flow of program execution (for example, how you arrived at a function, etc.).

This segment concludes the Software Analyzer tutorial.

2

Using the Software Analyzer

Using the Software Analyzer

This chapter provides an overview of the process of using the Software Analyzer. It then describes the tasks you perform when you use the Software Analyzer.

The overview includes information on the following:

- Microprocessor interface
- Inverse assembler
- Symbol file format
- Access to source code files
- Logic analyzer issues

The tasks include:

- Generate symbol files
- Start and exit the software analyzer
- Download symbols to the logic analyzer
- Make trace measurements
- View trace measurement results

Software Analyzer Overview

The HP Software Analyzer works with many microprocessors and within many embedded software development environments. It is a post-processing software application that correlates the logic analyzer's address label with a line of high level source code whose address, symbol name, file name and line number are described in a symbol file referenced by the logic analyzer.

The overall effectiveness of the Software Analyzer will vary to some degree depending on the specific development environment it is being used in. The following areas affect the performance of the Software Analyzer for different development environments.

Microprocessor interface All the information needed to reconstruct the complete address bus of the target system must be acquired by the logic analyzer.

Inverse assembler The logic analyzer's inverse assembler may need to reconstruct any incomplete address bus information and/or filter out any unexecuted instructions.

Symbol file format The compiler needs to produce a symbol file format that is readable by the Software Analyzer or a general purpose ASCII file format needs to be generated.

Access to source code files The logic analyzer must be able to access the high level source code files referenced by the Software Analyzer application.

Logic analyzer issues Logic analyzers have a number of inherent limitations that you need to be aware of as you use the Software Analyzer application.

Validation Process

HP has performed a validation process and written processor-specific application notes for some of the more popular embedded microprocessors. The microprocessor-specific application notes provide you with additional information to help you be successful with the Software Analyzer product. These notes are available at the ftp site ftp://col.hp.com/dist/swa.

Typically, with any given microprocessor you will also have better results with the software analyzer if the following two conditions are met:

- The processor has a microprocessor interface and inverse assembler developed by HP or one of its third-party partners. These microprocessors are listed in the publication, *Microprocessors and Bus Interfaces for HP Logic Analyzers*.
- The compiler being used generates a symbol file format that is currently supported by the HP Software Analyzer. A symbol file format converter to one of the supported formats would also be acceptable, provided the format specification for the supported format is followed. Converters often lose some symbol information. In these cases, product performance may be limited.

See Also

The supported file formats are listed in chapter 3, "Supported Symbol File Formats".

Microprocessor interface

The HP Software Analyzer needs a complete and accurate acquisition of the target system's address bus so that software symbols and source code line numbers can be correlated to the actual microprocessor code execution. The complexities of modern microprocessors can sometimes make complete and accurate address bus acquisition difficult.

Microprocessors with chip selects that are multiplexed with address lines, and target systems that utilize paged memory, can cause logical addresses in the program to be different than the physical addresses seen on the bus. The symbol tools require that there exist a way to map physical addresses one-to-one to the logical addresses.

For processors that use chip selects (or MMUs, paged memory, or reassigned address lines), the one-to-one mapping is not always possible. Therefore, analysis is required to determine if a mapping can be found between physical and logical addresses. This element requires the most target knowledge and is a key challenge for the designer using the Software Analyzer. It is discussed in detail in the processor-specific Motorola 68332 application note.

If you are designing your own microprocessor interface, exercise caution to ensure that a complete and accurate address bus is acquired by the logic analyzer. A custom microprocessor interface must clock the logic analyzer when the address bus is valid, properly acquire multiplexed address/data buses on the same state, and not clock the logic analyzer too frequently so that the same instruction is referenced multiple times by the Software Analyzer. The *HP E2445A User Definable Preprocessor Interface User's Guide* provides information and materials to help in accomplishing these tasks.

For complex microprocessors that utilize prefetching and bursting, a conversion of the processor bus to a full address bus (that represents the target system's software execution) is not always possible via hardware. Inverse assembly (post-acquisition software processing) is often required to achieve this. In these situations triggering on software address execution can be complicated to a degree. The logic analyzer triggering capabilities can provide triggering solutions for some of these situations. Two of these capabilities, Offset and Alignment, are discussed in the next two sections.

Using the Logic Analyzer Trigger Offset

Prefetching microprocessors will queue multiple instructions into an instruction buffer. If the function call or branch is taken, the microprocessor interface does not filter out these prefetched instructions. This means that the prefetches will be collected by the analyzer, and that a trigger set to one of these address will cause a false trigger. A common example of this is setting a trigger on the source line following a function call.

In the sample trace list below, lines 66 and 68 are prefetched following the function call to write_hdwr at line 63. So, a trigger set to line 66 or 68 will result in a premature trigger.

```
 Line  C Source Code
------  ----------------------------------------
   50    update_system(int passes)
   51    {
  .
  .
   62    /* Update hdwr_encode value.*/
   63    write_hdwr(func, hdwr_encode);
   64
   65    /* Save temp and humidity.*/
   66    save_points();
   67
   68    }
```

The recommended way to avoid false triggering due to prefetches is to offset the trigger address by the prefetch instruction queue depth. An offset field is provided in the trigger menu to allow offsetting the symbol address.

This is not a foolproof scheme for a couple of reasons. For one, using an offset of the full prefetch queue depth may result in a missed trigger if a branch takes place between the base address and the offset address. In this case, the queue depth offset is too large. Secondly, for processors that use instruction pipelining, the offset would need to cover the prefetched instructions in the queue and pipeline. In this case, the queue depth may not be large enough.

An algorithm for finding a more accurate offset value is listed below. (If the instruction length is less than the bus width, see "Using the Logic Analyzer Trigger Alignment.")

To calculate the offset value for an 'N'-bit bus/instruction set:

1 **Set a trigger with no offset on the line number desired.**

2 **If the trigger halted on a prefetched instruction, count the number of prefetched assembly instructions following (and including) the triggered instruction.**

3 **Determine the instruction length and bus width.**

For example, if the instruction length and bus width are 32 bits wide: one instruction occupies the width of the data bus and the address is incremented by four each time, since memory is organized in bytes (8 bits). Therefore, multiply the number found in step 2 by four.

Bus Width/ Instruction Length	**Multiply By:**
64 bits	8
32 bits	4
16 bits	2

4 **Convert this number to hexadecimal and place in the offset field of the trigger menu.**

Using the Logic Analyzer Trigger Alignment

When microprocessors utilize burst transactions, the address on the bus is held constant while several "bursts" of data are transferred to or from memory. Triggering on an individual address within the burst is not possible unless it is the first address (which is held on the bus). For this reason, an alignment should be specified in order to avoid missed triggers.

Another case when you need trigger alignment is when microprocessors have a wider data bus than instruction length. When this is the case, multiple instructions can be fetched on the data bus at the same time. For example, a 32-bit bus can retrieve two 16-bit long instructions during a given fetch. If two instructions are fetched, then the address for one of these instructions will be seen on the address bus while the other will not. This means a trigger set for the instruction not seen on the bus will be missed.

Source Line	C Source	Address
5	h = 0;	00000104
6	i = 1;	00000106
7	j = h + i;	00000108

In the above example, instruction fetches will occur at addresses 104 and 108, so that a trigger set on line 6 (address 106) will not occur! The instruction at address 106 was actually fetched with the 32-bit memory fetch at address 104, so we needed to trigger on address 104 to catch the fetch of 106.

To avoid these missed triggers, set a trigger alignment. This aligns the address to a specified boundary. Some examples include 1-, 2-, 4-, and 8-byte alignment. Alignment affects the least significant address bits of the trigger specification, either setting them to a don’t care or zero value.

For the example above, a 4-byte alignment should be set. Four-byte alignment will change the least significant two bits ($2^2 = 4$) of the trigger. Entering address 106 with four-byte alignment results in a trigger address range of 104 through 107 (2 bits of don’t care), or an address of 104 (2 bits of "0"). Note that either of these triggers would catch line 6.

Another way to accomplish the same result is to manually insert the address in the trigger field with the X’s or 0’s in place of the don’t care bits.

See Also

Trigger alignment is discussed in more detail in the Motorola 68040 or Intel 80186 Symbol Utility/Software Analyzer application notes, available from ftp://col.hp.com/dist/swa.

Inverse assembler

The HP Software Analyzer needs an accurate representation of the target system's program counter in order to correlate the logic analyzer execution trace to the corresponding source code that generated it. Transforming the address bus data acquired from the logic analyzer and microprocessor interface into a representation of the target system's program counter usually requires a software inverse assembler for most modern processors. Again, the architectural features of modern processors can make this software post-processing transformation difficult.

Prefetching To improve performance, most modern microprocessors prefetch instructions that may not actually be executed if the program branches. Even with normal inverse assembly, the Software Analyzer will correlate these unexecuted prefetches to their corresponding source code. Without some care, this can lead you to believe that sections of code were entered that were never really executed. HP has developed an advanced inverse assembler technology that allows you to filter out these unexecuted prefetches. At this time, this advanced inverse assembler technology is only available for select microprocessors. Most of these microprocessors have already been validated for the Software Analyzer by HP. For general purpose application of the Software Analyzer, you need to carefully identify unexecuted prefetches on your own.

Bursting Many processors utilize burst transactions. Most of the HP inverse assemblers compensate for burst mode by reconstructing the burst address from the address seen on the bus. If the inverse assembler does not compensate for burst mode, bursting should be disabled or the Software Analyzer will not properly correlate to instructions fetched during a burst cycle.

Missing least significant address bits Some processors encode the least significant bits of the address bus in information transmitted on status lines. Most of the HP inverse assemblers compensate for missing least significant address bits by reconstructing them from captured status information. If the inverse assembler does not compensate for missing least significant address bits, the Software Analyzer will not properly correlate to target system execution.

Most of the HP microprocessor interfaces come with an inverse assembler. Make sure that the inverse assembler is being used and that it is the correct inverse assembler for your target system configuration (processor variant, endian setting, etc.). The microprocessor interface documentation can help you check this.

The logic analyzer must capture address information under the ADDR label. This is the only logic analyzer acquisition label the Software Analyzer uses.

If an inverse assembler is not available from HP or one of its third party partners, an inverse assembler development package is available with the HP 10391B. Note that the creation of a custom inverse assembler for a complex microprocessor is a significant design task.

Symbol file format

The HP Software Analyzer needs to read the source code symbols. The source code symbols can be output to a file by the software compiler. A number of symbol file standards exist in the industry to facilitate this. Still, there are some issues you should be aware of.

- You must invoke the required compiler options to generate a symbol file with address, source file names, and source line numbers. The compiler documentation or compiler vendor can describe how to invoke these compiler options, as well as the option to specify the symbol file format generated.
- The HP Software Analyzer can read many of the most popular symbol file formats. If your compiler generates a symbol file format that is supported, the Software Analyzer will read the source code symbols.
- Many compiler vendors provide file format converters. Users should check for a converter to a file format HP supports with the Software Analyzer.
- Unfortunately, different compiler vendors sometimes interpret symbol file standards differently or unknowingly violate the standard. Luckily, this is a fairly rare occurrence. HP will make a best effort to resolve such conflicts between the HP symbol readers and the compiler vendor file formats. However, customers using compilers/file format combinations unvalidated by HP do so at their own risk.
- The HP symbol file readers may not demangle C++ symbols. Mangled C++ symbols are available for use and the trace listing will still correctly correlate to the appropriate source file name and line number.
- The HP E2450A Symbol Utility ships standard with the HP 1660C-series, HP 1670-series, and HP 16500C logic analyzers. However, it needs to be installed on the logic analyzer hard disk to use its readers in conjunction with the B3740A Software Analyzer.
- Older HP 1660-series logic analyzers and HP 16500A/B logic analyzer mainframes may have insufficient CPU memory for building large symbol tables. The symbol file building capability of a given logic analyzer varies depending on the file format and symbol content of the software. Maximum symbol file sizes for each logic analyzer are:
  HP 1660 and HP 1670 series = 512k to 1M
  HP 16500C = 1M to 3M.

- Customers who have a large code set should use an HP 1670D-series benchtop logic analyzer or an HP 16500C logic analysis system because of the greater CPU memory size of those models.
- The HP Software Analyzer also supports a general purpose ASCII file format. Users can write programs to generate this ASCII file format from their compiler's symbol file output or can enter the symbol information manually.
- Generating symbol database files for large symbol files can be time consuming on the logic analyzer. The nsbuild utility, provided with the Software Analyzer, can be used to generate symbol database files on the host computer.

See Also

For information on nsbuild, see "Generating Symbol Files" in this chapter.

Access to source code files

The HP Software Analyzer must be able to access the high level source code files referenced by the symbol information so that these source files can be displayed next to and correlated with the logic analyzer's execution trace acquisition. This requires you to be aware of a number of issues.

- Verify that the correct file search paths for the source code have been entered into the Software Analyzer. The B3740A Software Analyzer can often read and access the correct source code file from information contained in the symbol file, if the source code files have not been moved since they were compiled. Refer to Settings→Source View in the Reference chapter.
- If source code files are being referenced across a network, the HP logic analyzer networking must be compatible with the user's network environment. HP logic analyzers currently support Ethernet networks running a TCP/IP protocol and support ftp, telnet, NFS client/server and X-Window client/server applications. Some PC networks may require extensions to the normal LAN protocols to support the TCP/IP protocol and/or these networking applications. Users should contact their LAN system administrators to help setup the logic analyzer on their network.
- If the source code files are under a source code or version control utility, check the file names and paths carefully. These utilities can change source code file paths and file names. If these names are changed from the information contained in the symbol file, the Software Analyzer will not be able to find the proper source code file. These version control utilities usually provide an "export" command that creates a set of source code files with unmodified names. The software analyzer can then be given the correct path to these files.

Logic analyzer issues

The HP Software Analyzer uses a logic analyzer to acquire a history of the target system's software execution. Logic analyzers have a number of requirements and limitations that a user of the Software Analyzer should be aware of.

- Since a C source code line typically generates around 10 lines of assembly code, the Software Analyzer should be used with a logic analyzer having deep acquisition memory to be most effective. The deep memory of the logic analyzer will capture a larger section of high level software execution. For deep memory logic analysis, HP recommends the HP 1670D-series benchtop logic analyzers, or the HP 16555D or HP 16556D modules for the HP 16500 Logic Analysis System.
- You need to adjust the logic analyzer trigger to compensate for processor prefetching or pipelining of instructions. The logic analyzer has a trigger offset field to help facilitate this. The trigger offset value should be great enough to overcome the combination of the prefetch and pipelining queue depth.
- The user needs to adjust the logic analyzer trigger to compensate for processors that fetch multiple instructions per word. The logic analyzer has a trigger alignment field to help facilitate this. The trigger alignment reduces the address resolution that is being used, allowing triggers on addresses that are not actually seen on the address bus. This method is also used for triggering on instruction addresses that are fetched during a burst.
- The processor's instruction cache needs to be disabled for the logic analyzer to capture a correct and complete history of software execution.
- The memory management units must be configured for only physical or direct address translation. This means that the physical addresses captured by the logic analyzer will map directly to the logical addresses of the source program.

- You need to adjust the Software Analyzer to compensate for relocatable code segments or memory management units that produce fixed code offsets. The logic analyzer has an address offset field to help facilitate this. Entering the appropriate address offset will cause the Software Analyzer to reference the correct symbol information for the relocatable or offset code.
- You should configure the logic analyzer's storage qualification capabilities to store only those cycles that correspond to software execution (non-idle, non-BDM, etc).
- The Software Analyzer can exhibit long responses to requests for the next source line if either the unexecuted prefetch filter is not enabled or if the current trace listing corresponds to code from a library that is not in the source code search path. Unexecuted prefetch filters should be enabled whenever possible and the logic analyzer storage qualification can be used to avoid capturing library code routines.
- A single C source code line will generate many assembly instructions. The Find in Trace feature will not find a given C source code line unless the first assembly instruction generated from the source line has been acquired by the logic analyzer.

See Also

For more information on using the logic analyzer trigger alignment or trigger offset, see "Microprocessor interface" in this chapter.

Generating Symbol Files

This section shows you how to:

- Compile and place symbols in the OMF file.
- Use the nsbuild utility to generate a symbol database file.
- Create General-Purpose ASCII symbol files.

To view program symbols in the Listing or Waveform menus of the modules in the logic analyzer, you need to create a symbol database file. There are two ways to create a symbol database file, and both begin with OMF (Object Module Format) symbol files. The first method is to load the symbol file into a logic analyzer, and have the Symbol Utility create the symbol database file. The second method is to use nsbuild on your computer and generate the symbol database file. This file is then loaded into the logic analyzer.

If your language tools cannot generate an OMF symbol file compatible with the Symbol Utility, and a converter to a supported format is not available, you can create a symbol file in the General-Purpose ASCII (GPA) file format.

To compile and place symbols in the OMF file

- **Use your language tool "generate debugging information" option to compile, assemble, and link your program.**

The "generate debugging information" option usually saves your program's symbols in an OMF file. Refer to your software development tool documentation for details on generating OMF files.

See Also

"Supported Symbol File Formats" in the Reference chapter for a list of the symbol file formats that are supported by the HP E2450A Symbol Utility.

To generate a symbol database file using nsbuild

The OMF file created in the previous section will be loaded into the logic analyzer. During the load process, the logic analyzer creates a symbol database file, to be used by the logic analyzer for symbol referencing. Generating symbol database files on the logic analyzer for large symbol files can be time consuming. The nsbuild utility is an alternative way to generate symbol database files. The nsbuild utility is available on HP 9000/700, SunOS, Solaris, Windows 95, and Windows NT. The utility is not available on Windows 3.1.

Symbol database files have the extension .ns. Using the nsbuild utility, the symbol database file (.ns) is generated on the host computer. The symbol database files can then be downloaded to the logic analyzer.

Symbol database files built on the logic analyzer are equivalent to files built using the nsbuild utility with the -c option. The -c option is the default.

The command syntax for nsbuild is

```
nsbuild [-V] [-?] [-c] [-t] <source file> [<.ns file>]
```

`-V`	prints version info
`-?`	prints usage message
`-c`	do not save stack variables and types (default)
`-t`	saves stack variables and types
`<source file>`	OMF file to read
`<.ns file>`	name of symbol database (.ns) file to create

Example

```
nsbuild main.x
```

This produces a symbol database file named main.ns from the OMF file main.x.

See Also

For more information on downloading symbols, see "Downloading Symbols to the Logic Analyzer" in this chapter.

To create General-Purpose ASCII symbol files

A General-Purpose ASCII (GPA) symbol file is a user-generated file that contains symbolic information. You can use GPA symbol files when Object Module Format (OMF) files do not contain symbolic information, when you wish to define symbols that are not in the OMF file, or when the HP E2450A Symbol Utility does not directly support the OMF of your language tools.

You can typically get symbol table information from a linker map file to create a General-Purpose ASCII (GPA) symbol file.

- **Use an ASCII text editor or other text processing tool to create or modify a General-Purpose ASCII symbol file.**

Each entry in the symbol file must consist of a symbol name followed by an address or address range.

While symbol names can be very long, the logic analyzer system only uses the first 16 characters.

The address or address range corresponding to a given symbol appears as a hexadecimal number. The address or address range must immediately follow the symbol name, appear on the same line, and be separated from the symbol name by one or more blank spaces or tabs. Ensure that address ranges are in the following format:

```
beginning address..ending address
```

Example

```
main     00001000..00001009
test     00001010..0000101F
var1     00001E22     #this is a variable
```

This example defines two symbols that correspond to address ranges and one point symbol that corresponds to a single address.

See Also

"General-Purpose ASCII Symbol File Format" in the Reference chapter for a complete description of the GPA format.

Starting and Exiting the Software Analyzer

This section shows you how to:

- Start the Software Analyzer.
- Connect to a logic analyzer.
- Exit the Software Analyzer.

To start the Software Analyzer interface

- **Click the "HP Software Analyzer" icon (Windows 3.1, Windows NT, or Windows 95) or type "swa &" on the command line (UNIX).**

You can also start the Software Analyzer from the command line on a personal computer running Windows NT or Windows 95. The "swa" command has the following syntax:

```
swa [-a <ip_addr>] [-s]
```

`-a <ip_addr>` Specifies the host name or equivalent Internet (IP) Address of the logic analyzer. Enter IP addresses in integer dot notation (for example, 15.6.253.113).

`-s` Starts the Software Analyzer without opening the Source code window.

This command is not available with Windows 3.1.

To connect to a logic analyzer

1 In the Software Analyzer window, choose the Settings→Connect... command.

Note that the Connect menu option appears "greyed out" if you are connected to a logic analyzer. If you want to connect to a different logic analyzer, exit the interface and start it again.

2 Enter the host name or Internet (IP) address of the logic analyzer in the "Logic Analyzer LAN Address" box.

3 If you are using the Software Analyzer on a personal computer running Windows 3.1, select the appropriate "Network Protocol".

4 Click the **`Connect`** button.

If you cannot connect to the logic analyzer, refer to "If you have LAN connection problems" in the "Common Problems" chapter.

To exit the Software Analyzer

- In the Software Analyzer window, choose the File→Exit command.

Downloading Symbols to the Logic Analyzer

This section shows you how to

- Load a symbol file.
- View or remove loaded symbol files.

This section assumes you have already installed the HP E2450A Symbol Utility on your logic analyzer.

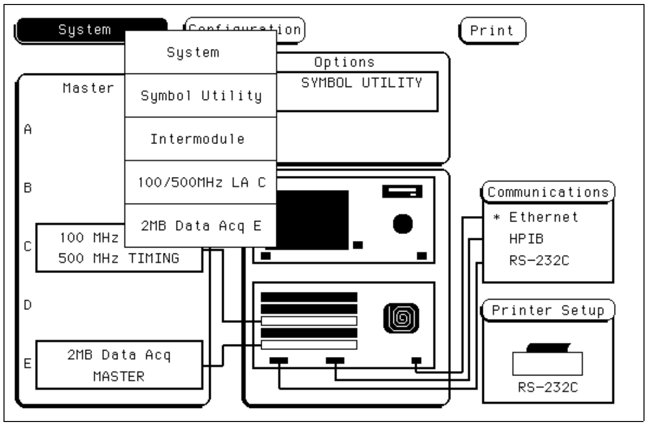
See Also

Instructions for installing the Symbol Utility software are in chapter 5, "Installation."

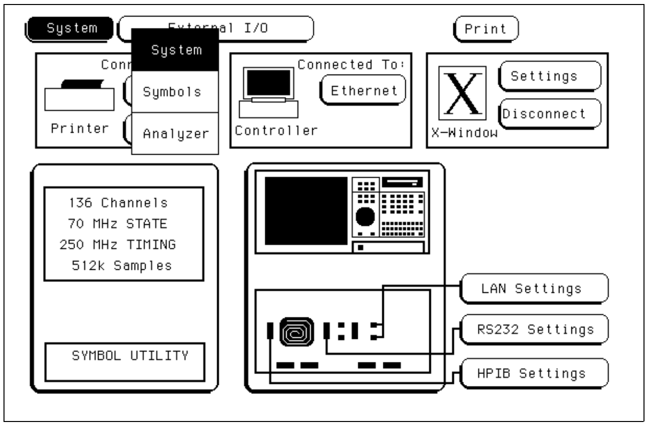
To load a symbol file

1 In the Software Analyzer window, choose the File→Download... command. Enter the local and logic analyzer file names, and click the **`Download`** button. The file from the host computer is copied to the logic analyzer.

2 In the Logic Analyzer window, select the Module field in the top left corner of the display. (When working with system-level menus, this field shows "System".) Choose **`Symbol Utility`** from the pop-up menu.

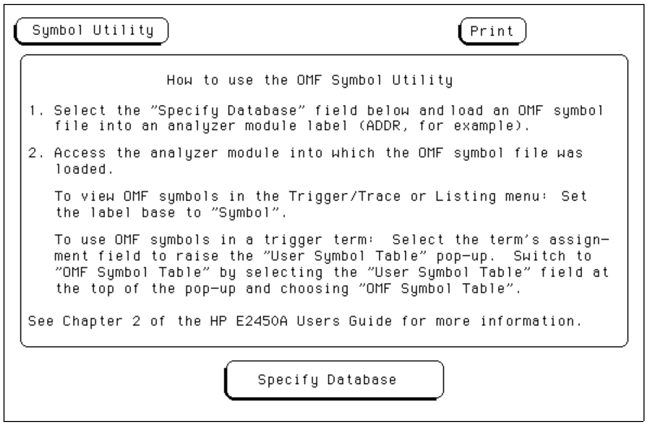


This figure shows the System Configuration menu of the HP 16500B.



This figure shows the System External I/O menu of the HP 1670A-series.

3 Select the **`Specify Database`** field in the Symbol Utility menu.



Symbol Utility

Print

How to use the OMF Symbol Utility

1. Select the "Specify Database" field below and load an OMF symbol file into an analyzer module label (ADDR, for example).

2. Access the analyzer module into which the OMF symbol file was loaded.

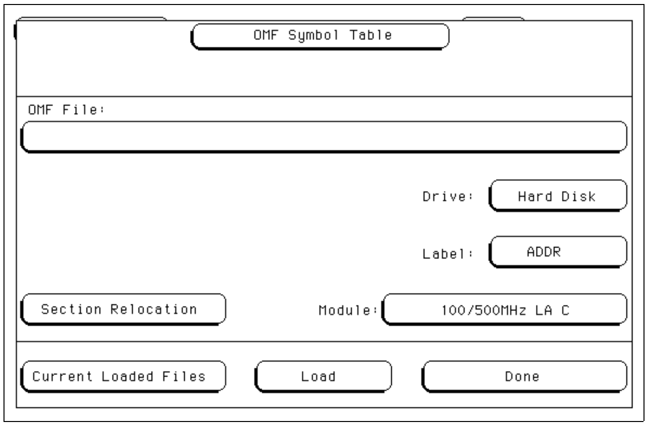
   To view OMF symbols in the Trigger/Trace or Listing menu: Set the label base to "Symbol".

   To use OMF symbols in a trigger term: Select the term's assignment field to raise the "User Symbol Table" pop-up. Switch to "OMF Symbol Table" by selecting the "User Symbol Table" field at the top of the pop-up and choosing "OMF Symbol Table".

See Chapter 2 of the HP E2450A Users Guide for more information.

Specify Database

4 Select the disk drive that contains the symbol file or the symbol database (.ns) file.



OMF Symbol Table

OMF File:

Drive: Hard Disk

Label: ADDR

Section Relocation

Module: 100/500MHz LA C

Current Loaded Files

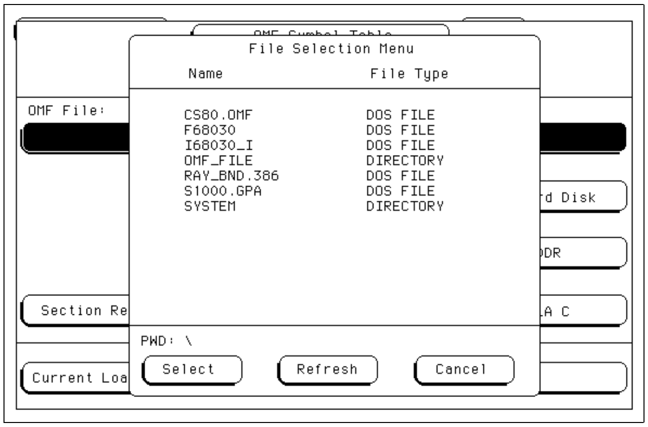
Load

Done

5 Select the logic analyzer module into which you want to load the symbol file. On benchtop logic analyzers, the module will always be **`Analyzer`**.

6 Select the **`Label`** field and choose the label to which you want to map the OMF symbols. (Typically, you will use the ADDR label for your system address bus.)

7 Select the **`OMF File`** field. In the pop-up menu, scroll to the desired file name. Click on the **`Select`** field to choose the file or change to a new directory. If you used the nsbuild utility to create a symbol database file on the host computer, select the database file (.ns) instead of the executable OMF file as the OMF File: selection.

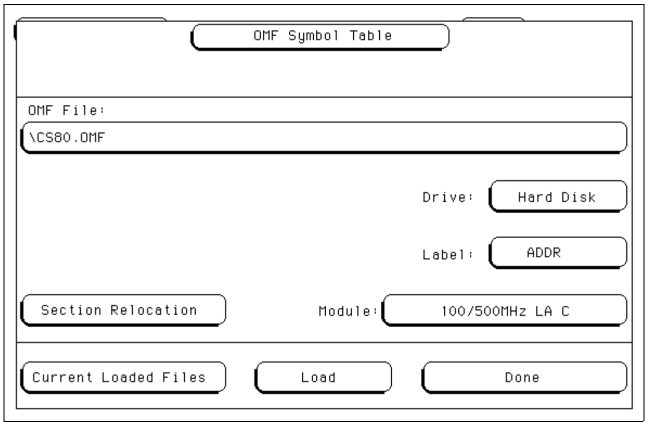


File Selection Menu

Name	File Type
CS80.OMF	DOS FILE
F68030	DOS FILE
I68030_I	DOS FILE
OMF_FILE	DIRECTORY
RAY_BND.386	DOS FILE
S1000.GPA	DOS FILE
SYSTEM	DIRECTORY

PWD: \

8 Select the **Load** field and wait for the database to be built.



OMF File: \CS80.OMF

Drive: Hard Disk

Label: ADDR

Module: 100/500MHz LA C

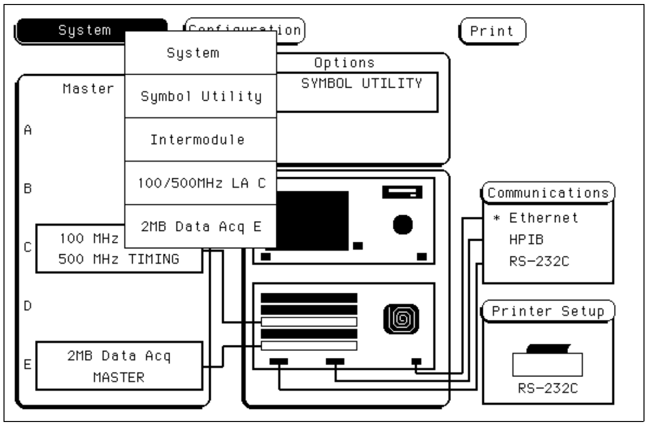
When you load a symbol file, the logic analyzer creates a database file. Database files have the extension ".ns". The database file appears in the same subdirectory as your OMF file.

The symbol database file loads into the measurement module you specified. You can load the same symbol database file into several modules, and you can also load several symbol database files into one module.

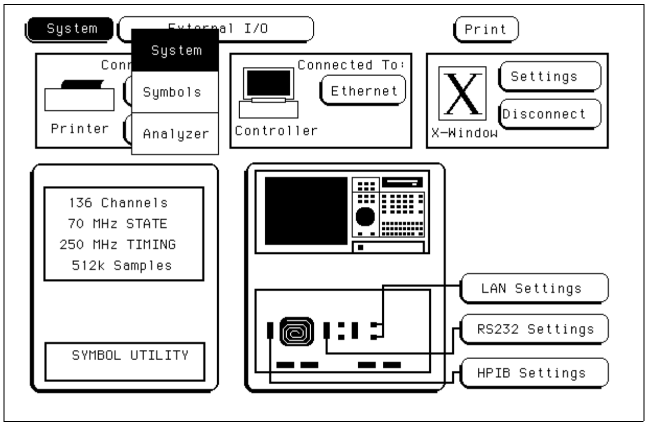
9 Select **Done**.

To view or remove loaded symbol files

1 In the Logic Analyzer window, select the module field in the top left corner of the display. Choose **`Symbols`** or **`Symbol Utility`** from the pop-up menu.

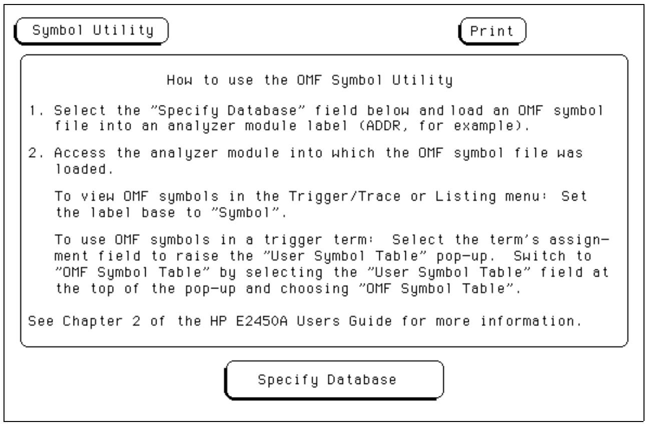


This figure shows the HP 16500B System Configuration menu.



This figure shows the HP 1670A-series System External I/O menu.

2 Select the **`Specify Database`** field in the Symbol Utility menu.



How to use the OMF Symbol Utility

1. Select the "Specify Database" field below and load an OMF symbol file into an analyzer module label (ADDR, for example).
2. Access the analyzer module into which the OMF symbol file was loaded.

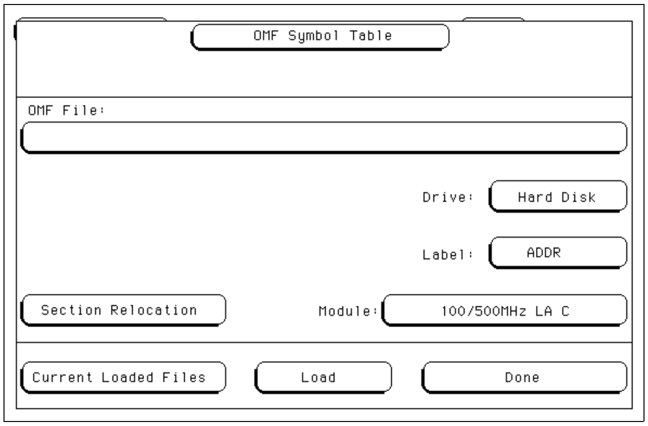
   To view OMF symbols in the Trigger/Trace or Listing menu: Set the label base to "Symbol".

   To use OMF symbols in a trigger term: Select the term's assignment field to raise the "User Symbol Table" pop-up. Switch to "OMF Symbol Table" by selecting the "User Symbol Table" field at the top of the pop-up and choosing "OMF Symbol Table".

See Chapter 2 of the HP E2450A Users Guide for more information.

This figure shows the HP 16500B version of the Software Analyzer. The version for benchtop logic analyzers is similar.

The Symbol Load menu appears.



OMF File:

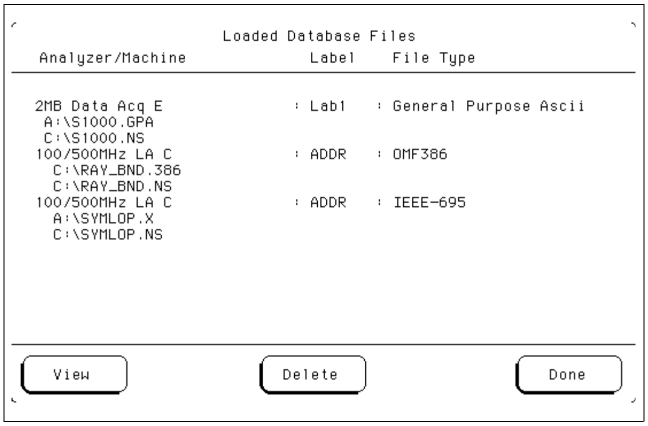
Drive: Hard Disk

Label: ADDR

Module: 100/500MHz LA C

3 Select the **`Current Loaded Files`** field in the bottom left corner of the display.

A list of currently loaded symbol database files is displayed.



```
                    Loaded Database Files
 Analyzer/Machine                Label    File Type

 2MB Data Acq E                : Lab1   : General Purpose Ascii
  A:\S1000.GPA
  C:\S1000.NS
 100/500MHz LA C               : ADDR   : OMF386
   C:\RAY_BND.386
   C:\RAY_BND.NS
 100/500MHz LA C               : ADDR   : IEEE-695
   A:\SYMLOP.X
   C:\SYMLOP.NS
```

4 To remove a symbol database file, highlight the module and file name that you want to delete. Then, select **`Delete`** to remove the symbol database file from the system.

5 To return to the OMF Symbol Table menu, select **`Done`**.

Making Trace Measurements

This section shows you how to make a trace measurement by triggering on a symbol.

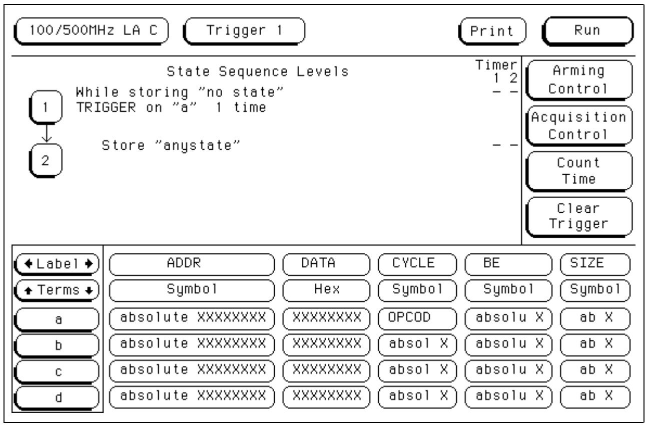
To trigger on a symbol

Before you can trigger on OMF symbols, you must load a symbol file into the logic analyzer, or build the symbol database file using the nsbuild utility and load the database file into the logic analyzer.

See Also "Downloading Symbols to the Logic Analyzer" for more information on loading files.

1 Go to the Trigger Menu.

2 Set the base of the label with which you want to specify a trigger term to **`Symbol`**. Typically, you use the ADDR label.



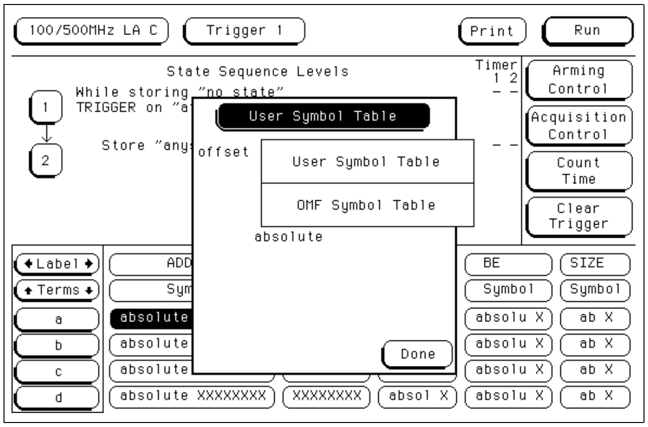
```
                State Sequence Levels              Timer
                                                    1 2
   While storing "no state"                         - -
   TRIGGER on "a"  1 time

      Store "anystate"                              - -
```

← Label →	ADDR	DATA	CYCLE	BE	SIZE
↑ Terms ↓	Symbol	Hex	Symbol	Symbol	Symbol
a	absolute XXXXXXXX	XXXXXXXX	OPCOD	absolu X	ab X
b	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X
c	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X
d	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X

3 Click on the assignment field of the trigger term that you want to use.

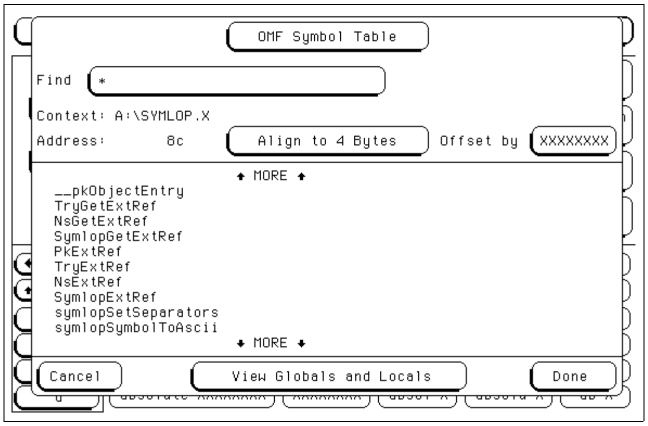
The assignment field is the field that corresponds to both the term column on the left side of the display and the label row in the center of the display.

4 In the pop-up menu, select the **`User Symbol Table`** field. Choose **`OMF Symbol Table`** from the pop-up menu.



5 Scroll through the list of symbols and pick the one that you want. Select **Done**.

The trigger term is now defined as one of your OMF symbols.



OMF Symbol Table

Find *

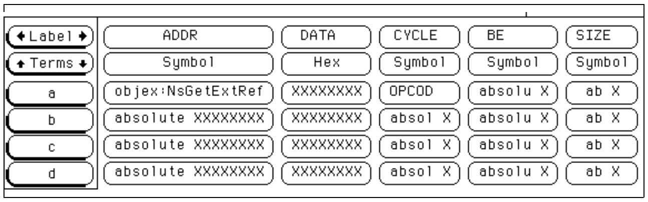
Context: A:\SYMLOP.X

Address: 8c   Align to 4 Bytes   Offset by XXXXXXXX

```
↑ MORE ↑
__pkObjectEntry
TryGetExtRef
NsGetExtRef
SymlopGetExtRef
PkExtRef
TryExtRef
NsExtRef
SymlopExtRef
symlopSetSeparators
symlopSymbolToAscii
↓ MORE ↓
```

Cancel   View Globals and Locals   Done

6 Use the trigger term that you just specified in the trigger specification to trigger the logic analyzer.



←Label→	ADDR	DATA	CYCLE	BE	SIZE
↑Terms↓	Symbol	Hex	Symbol	Symbol	Symbol
a	objex:NsGetExtRef	XXXXXXXX	OPCOD	absolu X	ab X
b	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X
c	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X
d	absolute XXXXXXXX	XXXXXXXX	absol X	absolu X	ab X

Viewing Trace Measurement Results

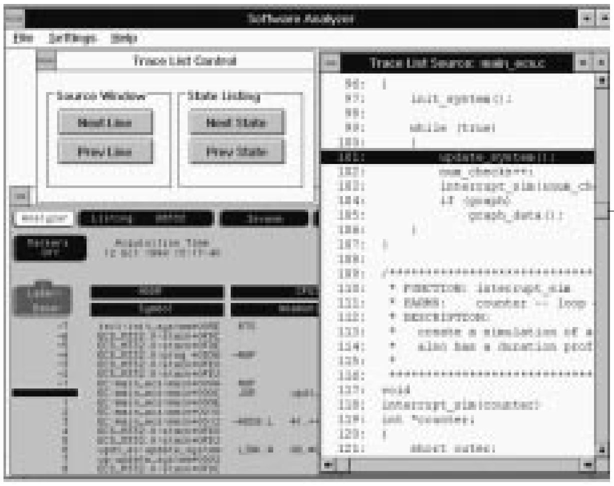
This section shows you how to:

- Scroll in the Source window.
- Scroll in the Logic Analyzer state listing window.
- Identify source file paths.
- Change Source window display options.
- Display symbols in the trace list.

To scroll in the Source window

- In the Trace List Control window, click the **`Next Line`** and **`Prev Line`** buttons.

- Highlighting a line in the state listing displays the corresponding source code in the Trace List Source window.



```
{
    init_system();

    while (true)
    {
        update_system();
        num_checks++;
        interrupt_sim(&num_ch
        if (graph)
            graph_data();
    }
}

/*****************************
 * FUNCTION: interrupt_sim
 * PARAMS:    counter -- loop
 * DESCRIPTION:
 *   create a simulation of a
 *   also has a duration prof
 *
 *****************************
void
interrupt_sim(counter)
int *counter;
{
    short sutes;
```

Clicking on the scroll bar, if present, lets you scroll through the source code—to view it only.

To scroll in the Logic Analyzer state listing window

Use one of the following procedures.

- In the Trace List Control window, click the `Next State` and `Prev State` buttons.

- In the Logic Analyzer window, with the state number field active, click the graphical knob buttons.

- In the Logic Analyzer window, with the state number field active, hold down the right mouse button and drag the mouse.

  Dragging up or to the right duplicates clockwise knob movement. Dragging down or to the left duplicates counter-clockwise knob movement.

- In the Logic Analyzer window, press the keyboard Shift-arrow keys.

  If you hold the Shift key down and press the up or right arrow key, you can duplicate clockwise knob movement. Pressing the down or left arrow key duplicates counter-clockwise knob movement.

To identify source file paths

1 In the Software Analyzer window, choose the Settings→Source View... command.

2 Type the source file path into the **`Source Path`** box.

3 If you want to insert the source file path before an existing path, in the list of paths, select the one that you want to insert first.

4 Click the **`Insert`** button.

5 When you finish inserting paths, click the **`OK`** button.

To delete a path from the list, select it and click the **`Delete`** button. A deleted path is automatically pasted into the **`Source Path`** box in case you want to re-insert it in a different place.

To change Source window display options

1 In the Software Analyzer window, choose the Settings→Source View... command.

2 To set the tabstops, enter the number of spaces between tabstops in the **`Tab Width`** box.

3 To show line numbers, select the **`Line Numbers`** check box.

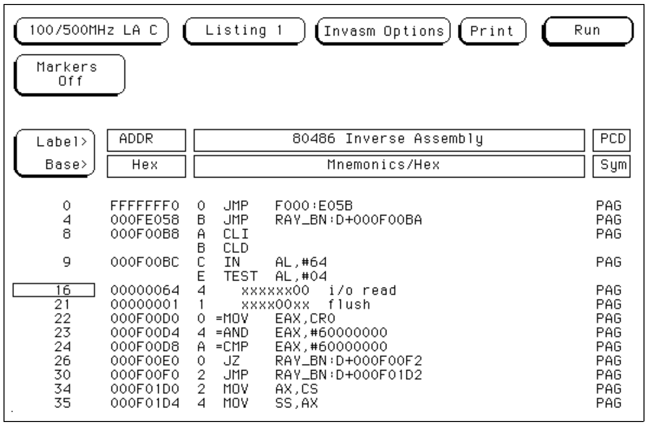
4 When you finish changing the Source window display options, click the **`OK`** button.

To display symbols in the trace list

Before you can display symbols in the trace list, you must load a symbol file or symbol database file into the logic analyzer.

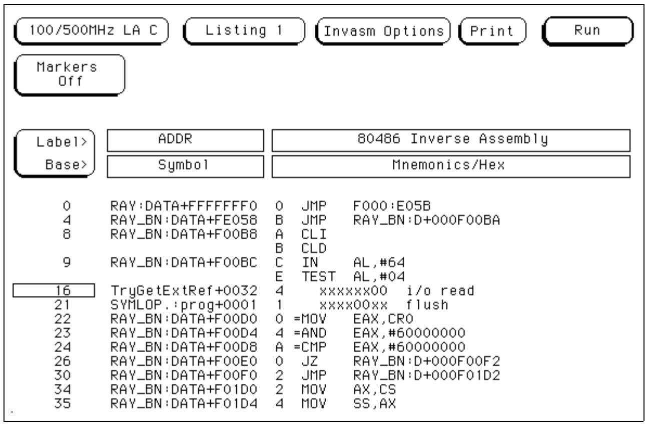
See Also "To load a symbol file" in this chapter for instructions on loading a symbol file.

1 Display the trace listing in the Analyzer Listing menu of the logic analyzer.



```
       ADDR        80486 Inverse Assembly                PCD
       Hex         Mnemonics/Hex                         Sym

    0  FFFFFFF0  0  JMP   F000:E05B                      PAG
    4  000FE058  B  JMP   RAY_BN:D+000F00BA              PAG
    8  000F00B8  A  CLI                                  PAG
                 B  CLD
    9  000F00BC  C  IN    AL,#64                         PAG
                 E  TEST  AL,#04
   16  00000064  4    xxxxxx00  i/o read                 PAG
   21  00000001  1    xxxx00xx  flush                    PAG
   22  000F00D0  0 =MOV   EAX,CR0                        PAG
   23  000F00D4  4 =AND   EAX,#60000000                  PAG
   24  000F00D8  A =CMP   EAX,#60000000                  PAG
   26  000F00E0  0  JZ    RAY_BN:D+000F00F2              PAG
   30  000F00F0  2  JMP   RAY_BN:D+000F01D2              PAG
   34  000F01D0  2  MOV   AX,CS                          PAG
   35  000F01D4  4  MOV   SS,AX                          PAG
```

2 Select the base of the ADDR label. If you loaded the symbols into a label other than ADDR, select the base for that label.



```
           ADDR                 80486 Inverse Assembly
           Symbol               Mnemonics/Hex

   0    RAY:DATA+FFFFFFF0  0  JMP   F000:E05B
   4    RAY_BN:DATA+FE058  B  JMP   RAY_BN:D+000F00BA
   8    RAY_BN:DATA+F00B8  A  CLI
                           B  CLD
   9    RAY_BN:DATA+F00BC  C  IN    AL,#64
                           E  TEST  AL,#04
  16    TryGetExtRef+0032  4    xxxxxx00  i/o read
  21    SYMLOP.:prog+0001  1    xxxx00xx  flush
  22    RAY_BN:DATA+F00D0  0 =MOV   EAX,CR0
  23    RAY_BN:DATA+F00D4  4 =AND   EAX,#60000000
  24    RAY_BN:DATA+F00D8  A =CMP   EAX,#60000000
  26    RAY_BN:DATA+F00E0  0  JZ    RAY_BN:D+000F00F2
  30    RAY_BN:DATA+F00F0  2  JMP   RAY_BN:D+000F01D2
  34    RAY_BN:DATA+F01D0  2  MOV   AX,CS
  35    RAY_BN:DATA+F01D4  4  MOV   SS,AX
```

3 Choose **`Symbol`** from the base pop-up menu field.

User Symbols take precedence over OMF symbols in the display. If you created User Symbols that overlap with OMF symbols, the User Symbols display in the listing instead of the OMF symbols.

3

Reference

Reference

This chapter describes the following:

- Main window commands
- Logic analyzer symbol file load menu
- Logic analyzer symbol browser menu
- Supported symbol file formats
- General-purpose ASCII symbol file format
- Messages

Main Window Commands

You can choose the following commands from the menu bar in the Main window:

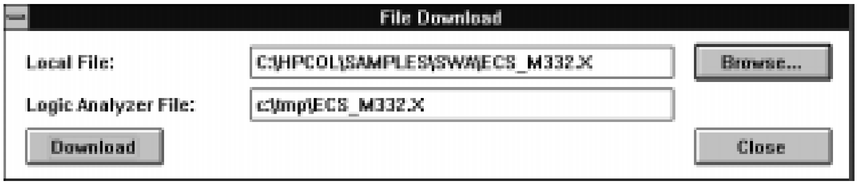
- File→Download...
- File→Exit
- Settings→Connect...
- Settings→Source View...
- Settings→Inverted Logic Analyzer Mouse (Sun only)

The Source File Lookup Failure dialog may appear while you use the Software Analyzer.

File→Download...

- **Select File→Download... to download files to the logic analyzer.**

These files are typically OMF files. You download them to build the symbol database on the logic analyzer. If you used the nsbuild utility, you can download the symbol database file (.ns file) instead of building the symbol database on the logic analyzer.



Local File: C:\HPCOL\SAMPLES\SWA\ECS_M332.X

Logic Analyzer File: c:\tmp\ECS_M332.X

`Local File` Specifies the local file for downloading to the logic analyzer.

`Browse...` Opens a dialog box that you use to select the local file.

`Logic Analyzer File` Specifies where the file is to be placed in the logic analyzer file system. Enter the logic analyzer path.

`Download` Starts the download and may bring up a "working" dialog which reports on the number of bytes transferred.

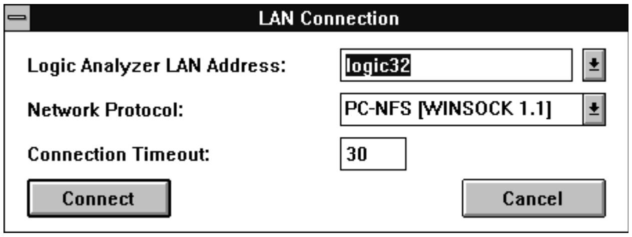
`Close` Aborts the file download and closes the window.

File→Exit

- **Select File→Exit to exit the Software Analyzer.**

Settings→Connect...

- Select Settings→Connect... to specify and make the LAN connection to the logic analyzer.



LAN Connection

Logic Analyzer LAN Address: logic32

Network Protocol: PC-NFS [WINSOCK 1.1]

Connection Timeout: 30

This dialog also appears when you do not specify the logic analyzer LAN address (and protocol) in the command line when starting the Software Analyzer interface.

`Logic Analyzer LAN Address` Specifies the host name or equivalent Internet (IP) Address of the logic analyzer. Enter IP addresses in integer dot notation (for example, 15.6.253.113).

`Network Protocol` *This option is present only on personal computers running MS Windows 3.1*, and lets you identify the protocol of your network. You can choose HP-ARPA, MS-TCP/IP [WINSOCK1.1], Novell-WP, or PC-NFS [WINSOCK1.1].

`Connection Timeout` Specifies the number of seconds to wait for the logic analyzer connection. Control of the interface returns when the connection is made or when the timeout occurs. Timeout values can be from 3 to 300 seconds.

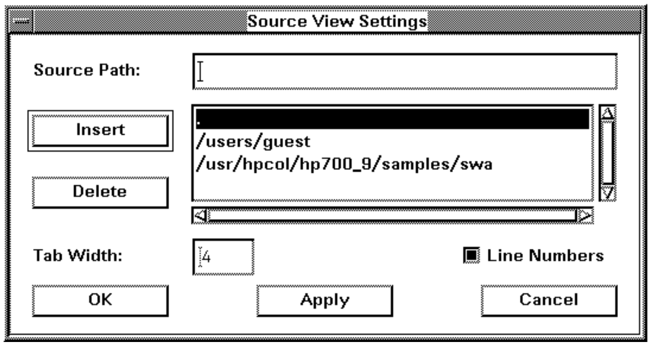
`Connect` Attempts the connection to the specified logic analyzer. If you cannot connect to the logic analyzer, refer to "If you have LAN connection problems" in the Common Problems chapter.

`Cancel` Closes the dialog box without connecting.

Settings→Source View...

- **Select Settings→Source View... to specify the source file search path and Source window settings.**

This figure shows the Source View Settings dialog box.



Source View Settings

Source Path:

.
/users/guest
/usr/hpcol/hp700_9/samples/swa

Tab Width: 4

■ Line Numbers

`Source Path` Lets you specify a source path to be inserted in the list. When looking for the source file to display in the Trace List Source window, the Software Analyzer searches the listed directories in the same order as they appear.

`Insert` Inserts the source path in the list, either before the highlighted line or at the end of the list if no line appears highlighted.

`Delete` Deletes the highlighted source path from the list. The deleted source path is copied to the entry box in case you want to insert it in a different place in the list.

`Tab Width` Sets the number of spaces between tab stops. Values are from 1 to 32.

`Line Numbers` Toggles line numbers ON or OFF.

`OK` Changes the settings and closes the dialog box.

`Apply` Applies the Tab Width and Line Numbers settings to the Trace List Source window without closing the dialog.

`Cancel` Closes the dialog box without changing the settings.

Settings→Inverted Logic Analyzer Mouse (Sun only)

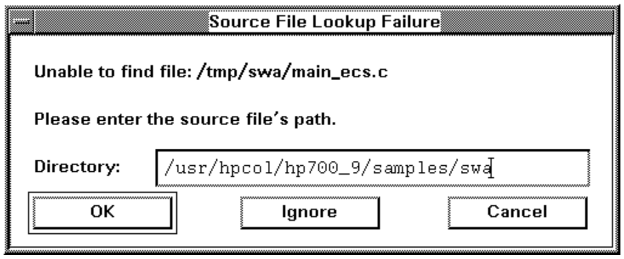
- **Select Settings→Inverted Logic Analyzer Mouse to toggle the use of the left and right mouse buttons in the Logic Analyzer window.**

In the default mode (not inverted), the mouse buttons behave as on the logic analyzer: left—click to activate a menu or button, right—drag to scroll a list.

If you turn on the "Inverted Logic Analyzer Mouse", the operations of the right and left mouse buttons are reversed (similar to the customary mouse button operation for Sun OpenWindows applications).

Source File Lookup Failure dialog box

This dialog box appears when the Software Analyzer cannot locate the source file for display in the Trace List Source window.



Source File Lookup Failure

Unable to find file: /tmp/swa/main_ecs.c

Please enter the source file's path.

Directory: /usr/hpcol/hp700_9/samples/swa

`Directory` Lets you enter the source path of the file that could not be found.

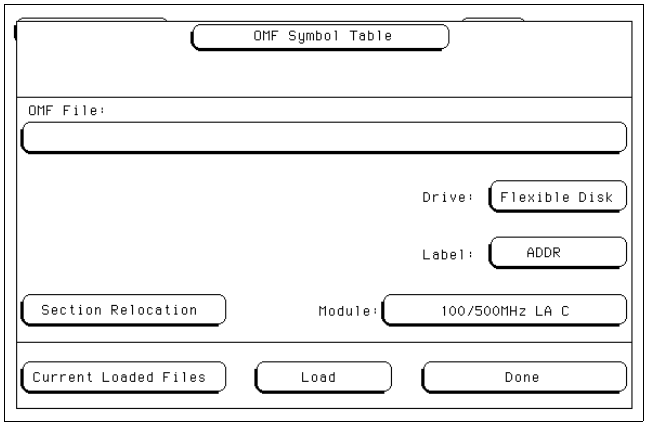
`OK` Looks in the specified directory for the source file. If the lookup succeeds, the new path is added to the list of paths in the Source View Settings dialog, and the Source window updates. An error dialog appears if the source file cannot be found.

`Ignore` No further attempt is made to find the source file for the rest of the session (unless its path is added to the Source View Settings dialog).

`Cancel` Ignores the source lookup; however, the dialog appears again the next time you attempt the source file lookup.

Logic Analyzer Symbol File Load Menu

The OMF Symbol Load menu loads the logic analyzer with the OMF files containing the desired symbols or the .ns files containing the pre-built symbol database.



Drive: Flexible Disk

Label: ADDR

Module: 100/500MHz LA C

The OMF Symbol Load menu areas that you can access are:

- OMF File field
- Drive field
- Label field
- Module field
- Current Loaded Files field
- Section Relocation option

OMF File field

Use the OMF File field to select the OMF symbol file or symbol database file that you want to load into the system. When you initially access the OMF Symbol Table menu, the OMF File field appears blank.

To use this field, select it using the mouse or keyboard. A File Selection pop-up menu appears. The pop-up menu shows you a list of files at the root of the drive currently selected in the Drive field. Scroll the list of files to select the symbol file that you want to use. Use the **`Select`** field to choose the file or to choose a subdirectory to browse. Select **`Cancel`** to close the menu.

Refresh field

The **`Refresh`** field causes the Symbol Utility to reread the contents of the disk drive. The Symbol Utility does not immediately read changes made to the disk drive contents. Use the **`Refresh`** key to reread the disk drive contents if they change.

Drive field

Use this field to specify the disk drive that contains the OMF file or symbol database file that you want to load. You can specify "Hard Disk" for the logic analyzer hard disk drive, or "Flexible Disk" for the logic analyzer flexible disk drive.

Label field

Use this field to specify the data label to which the symbols will correspond. In most cases, you will use the ADDR label because you will load symbols that correspond to processor address values. If you load symbols that correspond to another label, select this field and choose the appropriate label from the pop-up menu.

Module field

Use this field to specify the measurement module into which you want to load the symbols. When you select this field, the pop-up menu lets you choose from measurement modules currently installed in the HP 16500. If you are using an HP 1660C/CS-series or HP 1670-series benchtop logic analyzer, you can only choose **`Analyzer`**. Once the symbol file is loaded, you can view symbols in the Listing and Trigger menus of the specified module. To view symbols in more than one module, specify a different module in the **`Module`** field and load the symbol file again.

Load field

Select this field to load the file into the logic analyzer. During the load process, a symbol database file with a .ns extension is created for each OMF symbol file you load.

If you load the OMF symbol file from the hard disk drive, the .ns database file will be created in the same subdirectory containing the OMF file. If you load the OMF file from the flexible disk drive, the directory path from the flexible disk drive is duplicated on the hard disk drive, and the .ns database file is created in the duplicated directory.

Once the .ns file is created, the Symbol Utility will use this file as its working symbol database. The next time you must load symbols into the system, you can load the .ns file explicitly by placing the .ns file name in the **`OMF File`** field. You can purge the OMF file from the disk after the .ns file is created.

OMF file versions

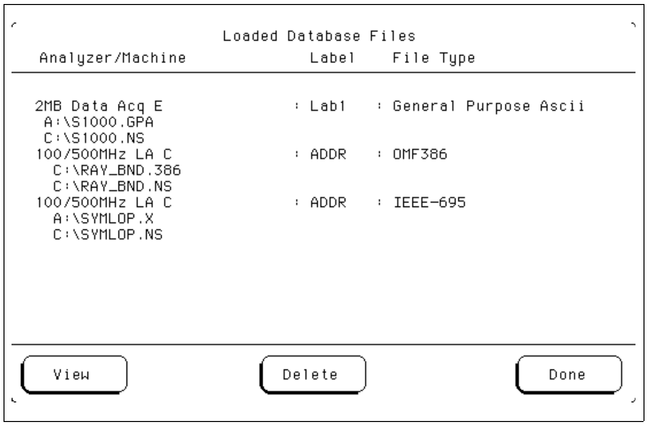
If you load an OMF file that was previously loaded, a new .ns file is created only if the OMF file is newer than the existing .ns file; otherwise, the existing .ns file is used.

Multiple files

You can load the same symbol database file into several modules, and you can load multiple symbol files into one module. Symbols from each of the files that you load will appear in the OMF Symbol Browser menu and can be used to create trigger terms for the logic analyzer.

Current Loaded Files field

Select this field to view a list of the symbol database files that are currently loaded into measurement modules.



```
                    Loaded Database Files
 Analyzer/Machine             Label    File Type

 2MB Data Acq E             : Lab1   : General Purpose Ascii
  A:\S1000.GPA
  C:\S1000.NS
 100/500MHz LA C            : ADDR   : OMF386
   C:\RAY_BND.386
   C:\RAY_BND.NS
 100/500MHz LA C            : ADDR   : IEEE-695
   A:\SYMLOP.X
   C:\SYMLOP.NS
```

Delete Field

The Loaded Database Files pop-up menu provides a **`Delete`** field that you can use to remove a symbol database from a measurement module. Highlight the module and file that you want to remove. Then, select the **`Delete`** field to remove the file.

View Field

The **`View`** field selects a symbol database in the OMF Symbol Table load menu. If more than one database has been loaded, you can choose the desired database by highlighting the analyzer and file name and selecting the **`View`** field.

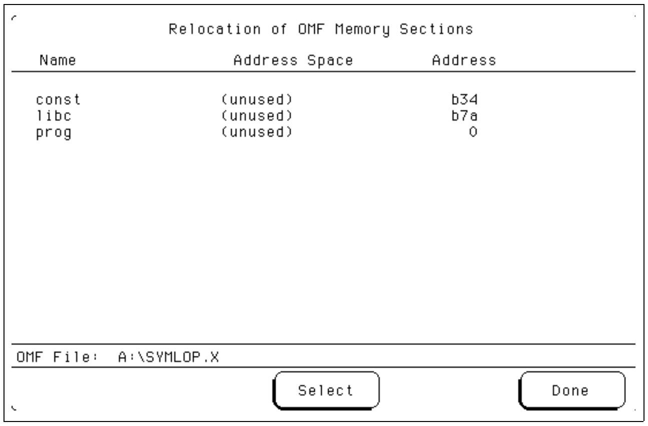
The file name that you select will appear in the **`OMF File`** field in the main OMF Symbol Table load menu. You can then change the file characteristics, such as the Section Relocation options.

Section Relocation option

The Section Relocation option lets you add offset values to the symbols in an OMF file. Use this option if some of the sections or segments of your code are relocated in memory at run time. Relocation can occur if your system dynamically loads parts of your code, causing the memory addresses where the code is loaded to not be fixed.

To use this option, highlight the memory section you want to offset and then click the **`Select`** field. Choose one of the offset options described below.

If you loaded multiple symbol files, the Section Relocation option applies to the file currently displayed in the **`OMF File`** field. To select a different file, use the **`View`** option in the Current Loaded Files menu.



```
                 Relocation of OMF Memory Sections
  Name                    Address Space         Address

  const                  (unused)                  b34
  libc                   (unused)                  b7a
  prog                   (unused)                    0

OMF File:  A:\SYMLOP.X
```

Set Absolute Section Location

When you know the run-time address of the section, use this option to set an absolute address for the start of the selected section.

Offset This Section

When you know the relocation offset of the section, use this option to add an offset to the start of the selected section. The value entered will be added to the section address that was contained in the OMF file. All of the symbols in the section's address range will be offset by this value.

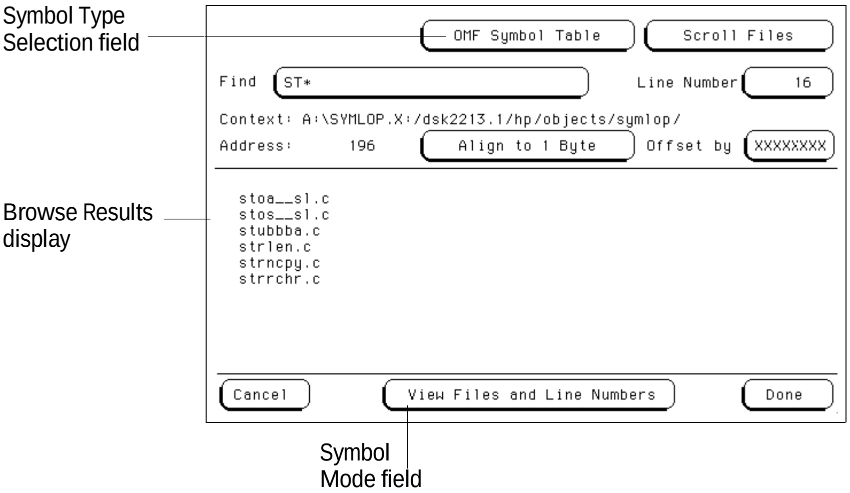
Offset All Sections

Use this option to add an offset to the starting addresses of all sections in the selected OMF file. The values entered will be added to all relocatable sections in your program. You will typically use this option if you have PC-relative code and data.

A section that contains symbols for hardware addresses will usually be indicated as nonrelocatable in the OMF file. The symbols for these sections will not be relocated.

Logic Analyzer Symbol Browser Menu

The OMF Symbol Browser menu lets you browse through the symbols that are loaded into a measurement module. You can use the symbols as trigger terms in the Trigger menu. Search features and wildcard characters are available to help you find the symbols that you want.



```
Find  ST*                                  Line Number  16
Context: A:\SYMLOP.X:/dsk2213.1/hp/objects/symlop/
Address:     196                           Offset by  XXXXXXXX

  stoa__sl.c
  stos__sl.c
  stubbba.c
  strlen.c
  strncpy.c
  strrchr.c
```

The OMF Symbol Browser areas you can access are:

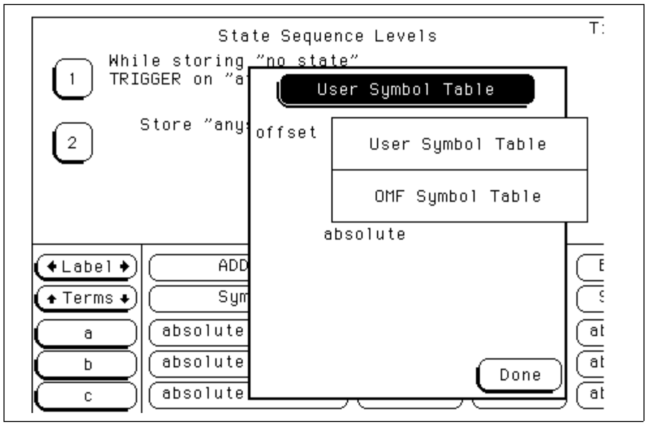
- Symbol Type Selection field (User or OMF)
- Find field
- Align to xx Byte option
- Offset option
- Browse Results display
- Symbol Mode field

Symbol Type Selection field (User vs. OMF)

This field lets you choose between the two types of symbols available in the logic analyzer:

- User Symbols correspond to the symbols that you can define in the Format menu.
- OMF Symbols correspond to the symbols in an OMF symbol file that you loaded using the Symbol Utility.

This field appears at the top of the pop-up menu when you select a trigger term.



Find field

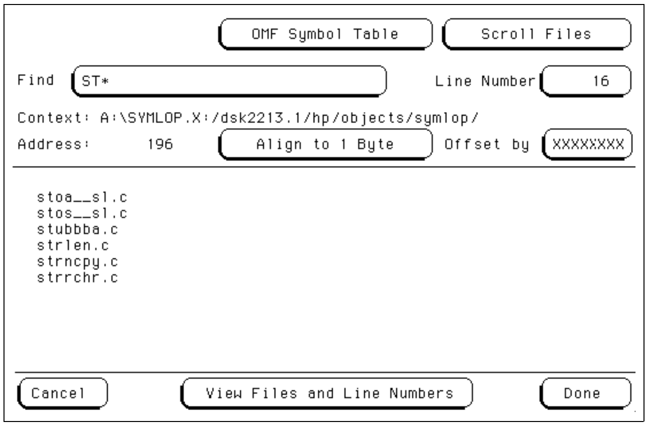
Use this field to locate particular symbols in the symbol databases that you want to use in a trigger specification. When you first access the OMF Symbol Table menu, the **Find** field will display an asterisk (*). Use the asterisk wildcard character to browse the entire symbol database for the symbol you want. To use this field, select it and type in the name of a file or symbol. Then, select **Done**. When you do this, the system searches the symbol database for symbols that match this name.

Asterisk wildcard (*)

The asterisk wildcard represents "any characters." When you search a symbol database using just the asterisk, you will see a list of all symbols in the database. You can also add an asterisk to a search word to find all symbols that begin or end with the same letters.

Example

To find all of the symbols that begin with the letters "st", select the **Find** field and type "st*". Select **Done** and the browse results will look like this:



```
Find ST*                                Line Number 16
Context: A:\SYMLOP.X:/dsk2213.1/hp/objects/symlop/
Address:      196                       Offset by XXXXXXXX

  stoa__sl.c
  stos__sl.c
  stubbba.c
  strlen.c
  strncpy.c
  strrchr.c
```

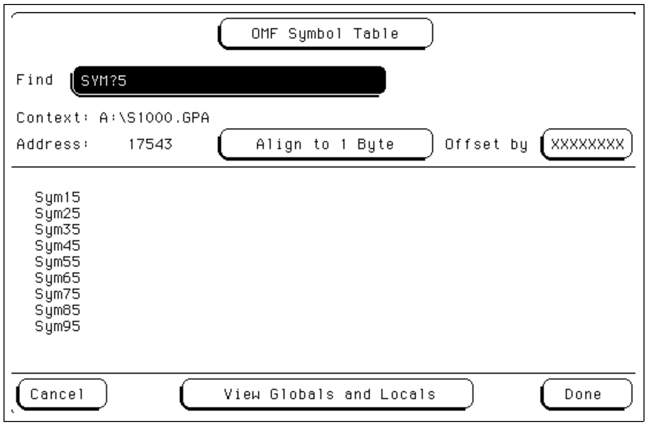
Now you can highlight the symbol that you want and select **Done** to use the symbol as a trigger term.

Question mark wildcard (?)

If you are using a keyboard to control your logic analyzer, you can use the question mark wildcard character. A question mark represents "any single character." You can use more than one question mark in a symbol database search; for example, "?ector?num" is a valid search string.

Example

Your symbol database contains many symbols that have names such as "sym1", "sym17", and "sym20". To find all of the symbols that end in "5", select the **Find** field and type "sym?5". Select **Done** and the browse results will look like this:



```
Find SYM?5

Context: A:\S1000.GPA
Address:    17543                              Offset by XXXXXXXX

  Sym15
  Sym25
  Sym35
  Sym45
  Sym55
  Sym65
  Sym75
  Sym85
  Sym95
```

Notice that "sym5" is not matched. Now you can highlight the symbol that you want and select **Done** to use the symbol as a trigger term.

Browse Results display

This part of the Symbol Browser menu lists the symbols that satisfy the search criteria specified in the **`Find`** field. You can scroll this field to view symbols that are off the screen.

Scroll Files field

This field appears if the symbol mode field, at the bottom of the screen, is set to "View Files and Line Numbers". Select this field to scroll the listing of the browse results. When the **`Scroll Files`** field is not selected, you can scroll the **`Line Number`** field.

Align to xx Byte option

Use the **`Align to Byte`** option to mask off an address. Most processors do not fetch instructions from memory on byte boundaries. To trigger a logic analyzer on a symbol at an odd-numbered address, you must mask off the address.

Example

The symbol "main" occurs at address 100F. The processor being probed is a 68040, which fetches instructions on long-word (4-byte) boundaries. To trigger on the address 100F, the address must be masked off to the nearest 4-byte boundary. The **`Align to Byte`** option would be set to "Align to 4 bytes". The Symbol Utility masks the address of the symbol "main" to 100C hex before it is used as a trigger term.

Offset option

Use the **`Offset`** option to add an offset value to the starting point of the symbol that you want to use as a trigger term. You might do this to trigger on a point in a function that is beyond the preamble of the function, or to trigger on a point that is past the prefetch depth of the processor. Setting an offset helps avoid false triggers in these situations.

The offset specified in the **`Offset by`** field is applied before the **`Align to xx Byte`** option masks the address.

Example

An 80386 processor has a prefetch depth of 16 bytes. Functions func1 and func2 are adjacent to each other in physical memory, with func2 following func1.

To trigger on func2, without getting a false trigger from a prefetch beyond the end of func1, add an offset value that is equal to or greater than the prefetch depth of the processor to your trigger term. In this case, you want to add an offset of 16 bytes, so you would set the value of the **`Offset by`** field to 10 hex. When you specify func2 as your trigger term, the logic analyzer will trigger on address func2+10.

Context display

The Context display, just below the **`Find`** field, indicates the original OMF source file for the symbol that is currently highlighted in the Browse Results display. "A:" indicates the logic analyzer flexible disk drive. "C:" indicates the logic analyzer hard disk drive. If the symbol database file was generated on the host computer using nsbuild, no drive designator will be present.

Some OMF formats, such as IEEE-695, provide information about the path name of the original source files. If you are using one of these OMF formats, you will see path information for the individual files in the browse results display. A colon separates the source file path from the OMF path by a colon.

Example

An OMF file called symlop.x was loaded from the hard disk drive of the logic analyzer. One of the source files for the OMF is called main.c. When you browse symbols from the symbol database, the context display might look like this:

```
Context: C:\symlop.x:/users/fred/project/src/main.c
```

C:\symlop.x is the path for the OMF file on the logic analyzer hard disk. The path /users/fred... applies to the file main.c in the original environment where it was compiled.

Address display

The Address display indicates the address corresponding to the symbol that is currently highlighted in the Browse Results display. The addresses are displayed as physical values. Intel 80x86 segment:offset values are converted to physical address values before they are displayed.

Symbol Mode field

You can view OMF symbols in one of two formats:

- Global and local variables
- Source file names with line numbers

Select the large field at the bottom of the menu to toggle between the two modes. Symbols appear in trace listings in the selected format.

When you select the **`View Globals and Locals`** mode, the browse results display lists symbolic names only. When you select the **`View Files and Line Numbers`** mode, the browse results display lists source file names from the OMF file.

In the **`View Files and Line Numbers`** mode, the following two additional fields appear in the OMF Symbol Browser menu.

Scroll Files field

Select this field to scroll the Browse Results listing.

Line Number field

Use this field to select a source file line number as a trigger term. Select the field once to scroll through the line numbers of valid, existing lines of code. Select the field a second time, and a keypad pop-up menu lets you specify a particular line number.

Not all lines in a source file have associated code. When you type in a line number that does not contain code, the field defaults to the next highest line number that does contain code. If you select a line number that is higher than any line number in the file, the field defaults to the highest line number in the file.

Supported Symbol File Formats

The Symbol Utility supports the following OMF file formats.

A.OUT This output format is used by the GNU tool chain for various processors, including the 68000 family. Other varieties of A.OUT are not supported.

ELF/DWARF This OMF is a portable format consisting of the ELF (Executable and Linkable Format) and the DWARF (Debugging Information Format) for various processors, including Intel 80960, PowerPC, and MIPS.

ELF/STABS This OMF is a portable format consisting of the ELF (Executable and Link Format) and the "stabs" debug format. The "stabs" format is derived from "The Stabs Debug Format," revision 2.125, produced by Cygnus Support. Deviations from this specification are not supported.

GPA This is the General-Purpose ASCII file format, which can be used to provide symbols for an Object Module Format that is not explicitly supported. See the "General-Purpose ASCII Symbol File Format" section in this chapter for more information.

HP-MRI IEEE-695 This OMF is produced by HP and MRI cross-compilers and cross-assemblers. This format is fully supported by the Symbol Utility IEEE-695 reader.

IEEE-695 This OMF is usually produced by language tools for non-Intel microprocessors. Some language tools that claim to output this format may deviate from the IEEE-695 standard in ways that make the OMF file unreadable by the Symbol Utility reader.

OMF286 This OMF is produced by language tools for Intel 80x86 series and Pentium microprocessors running in real and/or protected mode. The supported OMF286 file types are *Single-task Loadable* and *Bootloadable*.

OMF386 This OMF is produced by language tools for Intel 80x86 series and Pentium microprocessors running in real and/or protected mode. The supported OMF386 file types are *Loadable* and *Bootloadable*.

OMF86 This OMF is produced by language tools for Intel 80x86 series and Pentium microprocessors running in real mode only.

OMF96 This OMF is produced by language tools for the Intel 80196 family of microprocessors.

TI-COFF This OMF is produced by language tools for Texas Instruments microprocessors.

General-Purpose ASCII Symbol File Format

The Symbol Utility supports a General-Purpose ASCII (GPA) symbol file format. If your software development tools do not produce one of the supported symbol file formats, you can create a GPA file. You can also use a GPA file to define additional symbols that may not be included in the supported symbol file.

Various kinds of symbols are defined in different records in the GPA file. Record headers are enclosed in square brackets; for example, [VARIABLES]. For a summary of GPA file records and associated symbol definition syntax, refer to the following GPA Record Format Summary.

For more detailed descriptions of GPA file records and associated symbol definition syntax, refer to these topics in this chapter:

- SECTIONS
- FUNCTIONS
- VARIABLES
- SOURCE LINES
- START ADDRESS
- Comments

GPA Record Format Summary

Format

```
[SECTIONS]
section_name  start..end  attribute

[FUNCTIONS]
func_name  start..end

[VARIABLES]
var_name   start [size]
var_name   start..end

[SOURCE LINES]
File: file_name
line#  address

[START ADDRESS]
address

#Comments
```

If no record header is specified, [VARIABLES] is assumed. Lines without a preceding header are assumed to be symbol definitions in one of the VARIABLES formats.

Example

This is an example GPA file that contains several different kinds of records:

```
[SECTIONS]
prog     00001000..0000101F
data     40002000..40009FFF
common   FFFF0000..FFFF1000

[FUNCTIONS]
main     00001000..00001009
test     00001010..0000101F

[VARIABLES]
total    40002000  4
value    40008000  4
```

```
[SOURCE LINES]
File: main.c
10       00001000
11       00001002
14       0000100A
22       0000101E

File: test.c
 5       00001010
 7       00001012
11       0000101A
```

SECTIONS

Format

```
[SECTIONS]
section_name  start..end  attribute
```

Use SECTIONS to define symbols for regions of memory, such as sections, segments, or classes.

`section_name` A symbol representing the name of the section.

`start` The first address of the section, in hexadecimal.

`end` The last address of the section, in hexadecimal.

`attribute` This is optional, and may be one of the following:

NORMAL (default)—The section is a normal, relocatable section, such as code or data.

NONRELOC—The section contains variables or code that cannot be relocated; this is an absolute segment.

Enable Section Relocation

To enable section relocation, section definitions must appear before any other definitions in the file.

Example

```
[SECTIONS]
prog          00001000..00001FFF
data          00002000..00003FFF
display_io    00008000..0000801F  NONRELOC
```

If you use section definitions in a GPA symbol file, any subsequent function or variable definitions must be within the address ranges of one of the defined sections. The Symbol Utility ignores functions and variables that are not within the range.

FUNCTIONS

Format

```
[FUNCTIONS]
func_name  start..end
```

Use FUNCTIONS to define symbols for program functions, procedures, or subroutines.

`func_name` A symbol representing the function name.

`start` The first address of the function, in hexadecimal.

`end` The last address of the function, in hexadecimal.

Example

```
[FUNCTIONS]
main     00001000..00001009
test     00001010..0000101F
```

VARIABLES

Format

```
[VARIABLES]
var_name   start [size]
var_name   start..end
```

You can specify symbols for variables either by using the address of the variable, the address and the size of the variable, or a range of addresses occupied by the variable. If you specify only the address of a variable, the size is assumed to be one byte.

`var_name` A symbol representing the variable name.

`start` The first address of the variable, in hexadecimal.

`end` The last address of the variable, in hexadecimal.

`size` This is optional, and indicates the size of the variable, in bytes, in decimal.

Example

```
[VARIABLES]
subtotal     40002000   4
total        40002004   4
data_array   40003000..4000302F
status_char  40002345
```

SOURCE LINES

Format

```
[SOURCE LINES]
File: file_name
line#  address
```

Use SOURCE LINES to associate addresses with lines in your source files.

`file_name` The name of a file.

`line#` The number of a line in the file, in decimal.

`address` The address of the source line, in hexadecimal.

Example

```
[SOURCE LINES]
File: main.c
10       00001000
11       00001002
14       0000100A
22       0000101E
```

START ADDRESS

Format

```
[START ADDRESS]
address
```

`address` The address of the program entry point, in hexadecimal.

Example

```
[START ADDRESS]
00001000
```

Comments

Format

```
#comment text
```

Use the # character to include comments in a file. The Symbol Utility ignores any text following the # character. You can put comments on a line alone or on the same line following a symbol entry.

Example

```
#This is a comment.
```

Messages

This section describes messages that can appear when using the Software Analyzer.

Logic Analyzer disk full

Before you can download the file, you must recover logic analyzer disk space by purging unneeded files.

Logic Analyzer is already connected to a remote host

Only one remote interface may control a logic analyzer at a time.

This error also occurs when someone is accessing the logic analyzer file system as the "control user," or using telnet to access the logic analyzer command parser socket. The Software Analyzer must access the logic analyzer as the control user; there can only be one control user at any time.

The logic analyzer file system can be accessed by multiple "data users" if read-only access is required. Refer to the appropriate logic analyzer User's Guide for more information.

Bad transport type: <transport> (PCs with Windows 3.1 only)

A bad transport name was specified with the Software Analyzer "-t" command line option. The "transport" type can be HP-ARPA, MS-TCP/IP, Novell-WP, or PC-NFS. The transport types can be specified in upper- or lower-case characters, and only a significant number of characters are required. For example, "m", "ms-tcp", and "MS-TCP/IP" are all equivalent.

Cannot allocate additional memory

The Software Analyzer cannot acquire the memory it needs.

Cannot open file: <file>

The specified local file could not be found.

Communications error: <description>

This is a fatal communications error. The description tells you more about the cause of the problem.

Communications timed out

The connection timeout elapsed before the logic analyzer accepted the connection. See "If you have LAN connection problems" in the Common Problems chapter.

Empty directory name

An invalid source directory name was given in the Source Lookup Failure dialog.

Lost LAN connection

The established LAN connection was lost during a session. This happens if the logic analyzer is powered off or if the network goes down.

Networking DLL file <dll_name> not found (PCs with Windows 3.1 only)

Check your network configuration and PATH environment variable. There may be a mismatch between your configured PC networking and the Network Protocol selected in the LAN Connection dialog box. Refer to "If you have LAN connection problems" in the Common Problems chapter for further information.

No ADDR label defined

The **`Next Line`** or **`Prev Line`** button was pressed in an analyzer without an ADDR label. The most common cause is that the preprocessor configuration is not loaded.

No destination file

You must specify a logic analyzer filename for the file download.

No internet address for <IP address>

Make sure the specified Internet Address is valid. If you are using the analyzer’s name, make sure it is found in the "hosts" file or by the name server.

No source file

You must specify a local file for the download.

No source references found

The **`Next Line`** or **`Prev Line`** button was pressed, and no source line references were found between the current trace data state and the end (or beginning) of the trace data.

Unable to load logic analyzer fonts

The logic analyzer fonts could not be loaded for one of these reasons.

- On a personal computer, the fonts were not accepted by the video driver. Verify whether a patch exists for the video driver.
- On a UNIX workstation, the fonts could not be found in the X server font path. See the Common Problems chapter.

In addition, alternate fonts could not be loaded.

Unable to load logic analyzer fonts - using alternate fonts

The logic analyzer fonts could not be loaded for the same reason as the previous message. An alternate font that allows the product to run was loaded, but it may display inappropriate characters on some logic analyzer menus.

X Server is X11R4, not X11R5; the Logic Analyzer window may not render properly (UNIX only)

Some older (X11R4) X servers have a variety of problems displaying the Logic Analyzer window, including displaying unusual colors, or only showing part of the screen. The Software Analyzer requires an X11R5 server.

4

Common Problems

Common Problems

This chapter describes common problems that can occur when using the Software Analyzer, and some recommended solutions. These topics are discussed:

- If your logic analyzer is "in use"
- If you have LAN connection problems
- If you have alternate or missing fonts (Motif)
- If there are no symbols in the Listing menu
- If there is no source code in the Trace List Source window
- If Trace List Source navigation operates illogically
- If source and listing line numbers are different
- If Next Line/Prev Line operations take a long time

If your logic analyzer is "in use"

When you try to connect to a logic analyzer that is being used by someone else, a dialog box will appear and report the user connected to it.

- If the user is "High-Speed@remote", the logic analyzer communication channel is set to High-Speed SCSI or 16505 instead of Ethernet.
- If the user is "HP-IB@remote", the logic analyzer communication channel is set to HP-IB instead of Ethernet.
- If the user is "user@<ip_address>", the logic analyzer is connected to another computer; <ip_address> is the LAN address of the connected computer.

□ Select **`Cancel`** to cancel the connection attempt. Select **`Unlock`** to disconnect the previous user from the logic analyzer. Select **`Connect`** to disconnect the previous user, then attempt your connection again.

Disconnect an Existing User

Select either "Unlock" or "Connect" to immediately disconnect the previous user from the logic analyzer.

If you have LAN connection problems

□ If you are using a personal computer, verify that your PC network is working correctly. Try these methods:

```
ping <another_computer_on_your_network>
```

```
ftp <another_computer_on_your_network>
```

Because the Windows network is required to run this product, verify your Windows network configuration. Most networking packages install a Windows network device driver in the SYSTEM.INI file. If this is not done correctly, DOS-mode applications may run successfully, but the product will still fail under Windows.

□ Verify your logic analyzer network configuration.

1 Try to "ping" the logic analyzer from your computer:

```
ping <hostname or IP_address>
```

2 If the logic analyzer does not respond to a "ping", you need to verify its internet (IP) address. To do this, look at the logic analyzer's LAN settings. On an HP 16500B, these are under Ethernet Communication in the System Configuration menu. On an HP 16500C, the settings are under LAN Settings in the System Configuration menu. On a benchtop logic analyzer, the settings are under LAN Settings in the System External I/O menu.

The Ethernet configuration options (and example values) are:

```
Ether Address: 080009622842 (not configurable)
Analyzer name: LP16500B
Internet Address (IP): 15.6.253.106
Advanced Configuration
   Gateway Address: 15.6.248.1
   File Timeout: 1.5 s
   Subnet Mask: 255.255.248.0 (set automatically)
```

□ If you are using a personal computer running Windows 3.1, verify your network with the list of supported networks.

In the LAN Connection dialog, you can select the following Network Protocols:

MS-TCP/IP [WINSOCK 1.1] — This is the Microsoft TCP/IP networking package which is distributed by Microsoft for Lan Manager and Windows For Workgroups.

PC-NFS [WINSOCK 1.1] — This is the SunSelect PC-NFS networking package. You must install the Windows portion of this package to run this product. See the *PC-NFS Installation Guide* for more information.

Novell-WP — This is the Novell Lan Workplace for DOS. It is the only supported Novell networking package.

Novell Netware is NOT supported.

HP-ARPA — This is the Hewlett-Packard TCP/IP networking package.

□ If your networking package is *not* on this list, but it supports the Windows Sockets (Open Interface for Network Programming under Microsoft Windows) Version 1.1, choose the MS-TCP/IP [WINSOCK 1.1] network protocol. Because Windows Sockets is an open interface, this product *does not* guarantee functionality with all packages that claim to be compatible with Windows Sockets. You may try to operate this product at your own risk.

If you have alternate or missing fonts (Motif)

If the logic analyzer fonts for X/Motif have not been properly set up for your X server, you will see one of the following warning messages when you connect to a logic analyzer:

```
Unable to load Logic Analyzer fonts - using alternate fonts
Unable to load Logic Analyzer fonts
```

In the first case, you should still see text, but some of the characters on the logic analyzer Format or Configuration screens will not be displayed correctly. In the second case, when working on a PC, you will see no text in the logic analyzer window at all. In the second case, when working in X-Windows, you may see some changes in the arrows and activity bars.

□ **If your X server runs on the same computer as the Software Analyzer application, set the font path using the "xset" command from your shell prompt:**

```
/usr/bin/X11/xset +fp <install_root>/<platform>/lib/gsbpp
```

For example, if you installed the Software Analyzer in the default location (/usr/hpcol) on an HP 9000/700 workstation (hp700_9), the command would be:

```
/usr/bin/X11/xset +fp /usr/hpcol/hp700_9/lib/gsbpp
```

If you are using an HP 9000/700 workstation with the HP VUE window manager, you may add the appropriate "xset" command to the /usr/vue/config/Xstartup shell script to set the font path automatically each time you start the X server. If you start X via the "x11start" script, you can place the "xset" command in your $HOME/.x11start file.

On Sun workstations, you should run the <install_root>/<platform>/configure script to set up the logic analyzer font files.

See Also

For more information on font paths, refer to the "Installing the Software Analyzer" for your operating system in the Installation chapter.

□ **If you are using an X terminal, or are using the Software Analyzer through a remote X server connection, read the following.**

It is important that the font path (and the fonts themselves) is available to the X server to which your display is connected, and not on the workstation where the Software Analyzer application is run.

You may either NFS-mount the workstation that has the logic analyzer fonts installed and issue an "xset" command to add that directory to your font path (see the previous checkbox), or copy the logic analyzer fonts onto your local file system and add them to your font path from there.

For example, on an HP 9000 Series 700 workstation, you could use the following commands:

```
[NFS-mount the file system where the Software Analyzer is
installed]

mkdir /users/guest/fonts
cd /users/guest/fonts
cp /nfs/remotehost/usr/hpcol/hp700_9/lib/gsbpp/*.scf .
/usr/bin/X11/mkfontdir
/usr/bin/X11/xset +fp /users/guest/fonts
```

If your X server still does not display the logic analyzer fonts, you may need to convert them for your particular X server. For example:

```
cd /users/guest/fonts
rm *.scf
cp /nfs/remotehost/usr/hpcol/hp700_9/lib/gsbpp/*.bdf .
/usr/bin/X11/bdftosnf sm165.bdf > sm165.snf
/usr/bin/X11/bdftosnf lg165.bdf > lg165.snf
/usr/bin/X11/mkfontdir
/usr/bin/X11/xset +fp /users/guest/fonts
```

If there are no symbols in the Listing menu

□ **Make sure the Symbol Utility is present.**

The HP E2450A Symbol Utility is shipped with the Software Analyzer.

- If the Symbol Utility is loaded on an HP 16500B/C logic analyzer, a green box appears in the "Options" area in the System Configuration menu. The System Configuration menu is the first menu you see when you power up the HP 16500.
- If the Symbol Utility is loaded on an HP 1660C/CS-series or HP 1670-series logic analyzer, a grey box appears in the bottom leftmost box in the External I/O menu. Go to the External I/O menu.

The logic analyzer has the ability to load only two options. If you see that two options are loaded and the Symbol Utility is not one of them, you must remove one of the options and reboot the logic analyzer.

1. **To remove an option, go to the System Hard Disk menu and change to the C:\SYSTEM directory.**
2. **Option boot files begin with "OPT_". The Symbol Utility boot file is OPT_032 for HP 16500B, OPT5_032 for HP 16500C and OPTC_032 for the HP 1660C/CS-series and HP 1670-series. Rename one of the other option boot files.**

□ **Make sure a symbol file is loaded into the desired label for the correct module and machine.**

The Symbol Utility can load a symbol file into any existing label of any module and machine present in the logic analyzer.

1. **To determine how the symbol file was loaded, go to the OMF Symbol Table menu in the Symbol Utility and select "Current Loaded Files".**
2. **You will see which symbol files have been loaded into which module/machine, and the label to which the symbols are mapped. This will usually be the ADDR label.**

If there is no source code in the Trace List Source window

To make the Trace List Source window track the Listing menu, you must take the following actions.

□ Make sure the Symbol Utility is present (see "If there are no symbols in the Listing menu" on the previous page).

□ Make sure an ADDR label is turned on in the module and machine.

The Software Analyzer Trace List Source window tracks the center state in the Listing menu according to the source line reference associated with the analyzer's ADDR label. The ADDR label must be present in the module and machine for which you want source line referencing to work.

1 To verify that an ADDR label is turned on, go to the Format menu and look for "ADDR" in the list of labels. If there is no ADDR label, you may not have loaded the correct configuration file.

2 Note that the ADDR label does not need to be present in the Listing menu. It just needs to exist in the Format menu. Also, if the ADDR label is present in the Listing menu, it can be displayed in any base — binary, hex, symbol, etc.

□ Make sure a symbol file is loaded into the ADDR label for the correct module and machine.

The Symbol Utility can load a symbol file into any existing label of any module and machine present in the logic analyzer.

1 To determine how the symbol file was loaded, go to the OMF Symbol Table menu in the Symbol Utility and select "Current Loaded Files".

2 Verify that the proper symbol file was loaded into the correct analyzer and machine, and that it was loaded into the ADDR label.

If you are using more than one analyzer module, it is very easy to load the symbol file into the wrong module by mistake. Before you select **Load**, make sure you have selected the correct module and machine.

□ Make sure the Software Analyzer is able to access the source code files.

The Trace List Source window attempts to display the source file that corresponds to the center state of the Listing menu. First, it looks in the location given by the OMF file. Then, it searches the list of directories in the Source View Settings dialog. If the file is not found, a dialog prompts you to enter the name of the directory that contains the source file.

□ Make sure the loaded symbol database file has source line information in it.

Source line information in an OMF file normally consists of a list of line numbers and corresponding addresses. The Software Analyzer uses this information to make the Trace List Source window track the center state in the Listing menu. You must compile and link the executable program with the appropriate debug options so that the resulting OMF file contains source line information.

An easy way to determine whether or not the symbol or symbol database file has source line information is to load it using the Symbol Utility.

1. Go to the Symbol Utility OMF Symbol Table menu and load the OMF symbol file or .ns file into a module.
2. Go to the Trigger menu for the module.
3. In the list of Terms, set the base of the label in which you loaded the file to "Symbol".
4. Select a term under that label and change to "OMF Symbol Table".
5. Change the view to "View Files and Line Numbers". If there are no files in the list, or if you scroll through the list and the Address for each file is "No Address", there is no source line information in the file. Check your compiler documentation to ensure that you compiled with the debug option turned on.

If Trace List Source navigation operates illogically

When scrolling the Listing menu or using the Trace List Control buttons, you may see the highlighted line in the Trace List Source window move in an illogical manner. For example, as you move through the listing, the highlighted line may go to a line with a function call, then to the line after the function call, and finally to the source code in the function itself. Or, as you move through the listing, the highlighted line goes over both branches of an if-then-else statement.

This behavior is caused by a microprocessor instruction prefetch. Many microprocessors perform instruction prefetch. Prefetch means the microprocessor reads several instructions from memory ahead of the instructions that are executing. Prefetched instructions may or may not be executed, depending on the flow of the program.

The disassembler in the logic analyzer attempts to determine which instructions have been executed and which have not. If the disassembler has state suppression capability, you can suppress the states containing unexecuted instructions by selecting the appropriate menu item in the Invasm Options menu. When you suppress these states, the code flow viewed in the Trace List Source window is much more reasonable.

However, often heuristic algorithms must be applied to determine which states contain unexecuted instructions. For some microprocessors, this is not perfectly reliable. Therefore, unexecuted instructions may still appear in the Listing even though these states are supposed to be suppressed (as specified in the Invasm Options menu).

If source and listing line numbers are different

Certain circumstances can cause the highlighted line in the Trace List Source window to not match the line number in the listing menu.

The file and line number shown under the ADDR label is a result of an address lookup using the exact address shown there. The address used to place the highlighted line in the Trace List Source window is a combination of the address under the ADDR label and any extra address bytes shown under the DATA label.

The extra address bytes under the DATA label are present only when the DATA label is displayed in the Invasm base and is present on the screen. These characters always appear in the first two columns in the DATA label.

By combining the extra address characters with the address shown in the ADDR label, a more precise address can be obtained. This derived address may result in a source line reference that differs from the raw address.

If Next Line/Prev Line operations take a long time

After selecting one of the Trace List Control buttons, several seconds to several minutes can elapse before the highlighted line in the Trace List Source window moves to the appropriate line.

This elapsed time can occur when several suppressed states have corresponding source lines references. If the suppressed states do not have source line references (for example, if they are data reads or writes) this problem probably will not occur. When states with source line references are suppressed, the logic analyzer takes much longer to reposition the Listing menu.

The amount of elapsed time depends on the disassembler used, the number of suppressed states with source line references, and the number of source line references in the symbol database.

5

Installation

Installation

This chapter shows you how to install the HP B3740A Software Analyzer and other required software. These topics are discussed:

- Installing the Software Analyzer (MS Windows/Windows NT/ Windows 95)
- Installing the Software Analyzer (HP 9000/700)
- Installing the Software Analyzer (SunOS)
- Installing the Software Analyzer (Solaris)
- Installing the Logic Analyzer Software
- Installing the Symbol Utility Software

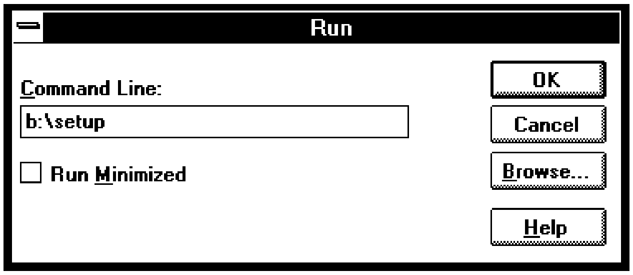
Installing the Software Analyzer (MS Windows/Windows NT/Windows 95)

To install the Software Analyzer on personal computers running Windows 3.1, you will take the following steps:

- Step 1. Install Win32s
- Step 2. Install the Software Analyzer
- Step 3. Set up the environment variables

To install the Software Analyzer on personal computers running Windows NT or Windows 95, you will follow the same process but skip step 1.

Step 1. Install Win32s



Command Line: b:\setup

□ Run Minimized

This step is only necessary if your PC is running Windows 3.1.

Before installing the Software Analyzer, you must install the Win32s libraries from the Microsoft Win32s Setup disks which are included with the Software Analyzer.

The Software Analyzer is a Microsoft Windows 32-bit application. To run 32-bit applications under MS Windows 3.1, the Win32s support libraries from Microsoft must be installed on your system.

The Software Analyzer requires Win32s version 1.1 or later.

1 Start MS Windows in the 386-enhanced mode.

2 Insert the Microsoft Win32s Setup disk into floppy disk drive A: or B:.

3 Choose the File→Run... command in the Windows Program Manager. Enter "a:\setup" (or "b:\setup" if you installed the floppy disk in drive B:) in the Command Line text box.

4 Click on the **OK** button. Then, follow the instructions on the screen.

The Win32s setup will ask you to add SHARE.EXE to your AUTOEXEC.BAT file. The Software Analyzer does not require you to run SHARE.EXE, but you may do so if you want.

Step 2. Install the Software Analyzer

1 Insert the Software Analyzer system disk in floppy disk drive A: or B:.

2 Choose the File→Run... command in the Windows Program Manager. Enter "a:\setup" (or "b:\setup" if you installed the floppy disk in drive B:) in the Command Line text box.

3 Click on the **OK** button. Then, follow the instructions on the screen.

Step 3. Set the environment variables

1 Set the PATH environment variable to include <dir>\bin where <dir> is the directory (including the drive designator) where you installed the software. For example, `set PATH=%PATH%; c:\hpcol\bin`.

2 Set the HPCOL_PRODUCTS environment variable to the directory where you installed the product. For example, `set HPCOL_PRODUCTS=C:\hpcol`.

After you have set the environment variables, you can start the Software Analyzer by double-clicking on the Software Analyzer icon.

Installing the Software Analyzer (HP 9000/700)

This section shows you how to install the Software Analyzer on HP 9000 Series 700 computers running the HP-UX operating system.

The Software Analyzer requires:

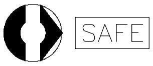
- 6 Mbytes of hard disk space
- HP-UX version 9.0 or greater
- Motif 1.2
- X11R5 X Window System or HP VUE

Refer to your HP-UX documentation for instructions on installing software (see the section on updating HP-UX).

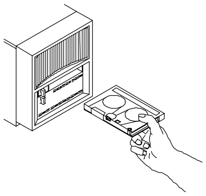
Step 1. Load the tape

1 **Become the root user on the system you want to update.**

2 **Make sure the tape write-protect screw points to SAFE.**



3 **Put the product media in the tape drive that will be the *source device* for the update process.**



4 Confirm that the tape drive BUSY and PROTECT lights are on. If the PROTECT light is not on, remove the tape and confirm the position of the write-protect screw. If the BUSY light is not on, check that the tape is installed correctly in the drive and that the drive is operating correctly.

Step 2. Install the Software Analyzer

1 For version 9.xx, complete the following steps.

a When the BUSY light goes off and stays off, at the HP-UX prompt start the update program by typing:

```
/etc/update
```

b When the HP-UX update utility main screen appears, confirm that the source and destination devices are correct for your system. Refer to your HP-UX documentation (see the section on updating HP-UX) if you need to modify these values.

c Choose "Select All Filesets on the Source Media" when your source and destination directories are correct.

d To begin the update, press the softkey <Select Item>. At the next menu, press the softkey <Select Item> again. Answer the last prompt with "y".

It takes several minutes to read the tape.

e When the installation is complete, read `/tmp/update.log` to see the results of the update.

f Skip the next step, and continue with Step 3: Set the PATH environment variable.

2 For version 10.xx, complete the following steps:

a Clear the permissions mask:

```
umask 0
```

b Change to the directory under which you want the software to be installed. For example:

```
cd /
```

All of the software will be installed into the subdirectory usr/hpcol.

c Extract your software. Type:

```
tar xvfbp <device> 20 2>&1 | tee tar.log
```

If you are installing the software from a tape drive which is on a remote system, type the following command (one line):

```
remsh <hostname> "dd if=<device> bs=20b" |
tar xvfbp - 20 2>&1 | tee tar.log
```

<device> is the name of your tape drive. <hostname> is the name of the system with the tape drive.

Step 3: Set the environment variables

1 To allow your X server to find the logic analyzer fonts, at the HP-UX prompt type:

```
/usr/bin/X11/xset +fp <dir>/usr/hpcol/hp700_9/lib/gsbpp
```

Where <dir> is the directory where you installed the software.

If you are using the HP VUE window manager, you may add the "xset" command to the /usr/vue/config/Xstartup script to set up the font path each time the X server is started. If you are not using VUE, you may add it to your $HOME/.profile file.

2 Set the PATH environment variable to include a <dir>/bin directory where <dir> is the directory where you installed the software. For example, `export PATH=$PATH:/usr/hpcol/hp700_9/bin`

3 Set the HPCOL_PRODUCTS environment variable to the directory where you installed the product. For example, `export HPCOL_PRODUCTS=/usr/hpcol/hp700_9`.

After you have set the environment variables, you can type "swa &" to start the Software Analyzer in background mode.

Installing the Software Analyzer (SunOS)

System Requirements

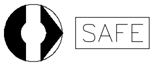
This software is designed to run on a Sun SPARCsystem with SunOS version 4.1.3 or 4.1.3_U1 and OpenWindows 3.0. The tape uses the QIC-24 data format. Your system must have 8 Mbytes of free disk space to store this software.

To install the Software Analyzer on Sun workstations running the SunOS operating system, you will take the following steps:

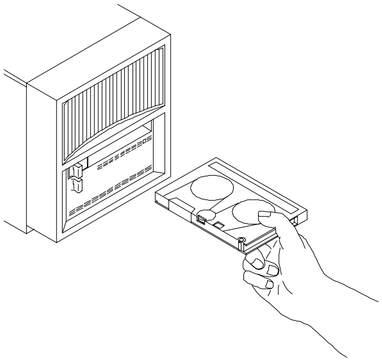
- Step 1. Load the tape.
- Step 2. Install the Software Analyzer.
- Step 3. Run the configure script (if necessary).
- Step 4. Install a SunOS patch.
- Step 5. Set the environment variables.

Step 1. Load the tape

1 Verify that the HP B3740A tape write-protect screw points to SAFE.



2 Insert the tape in the tape drive, and slide the lever on the tape drive to the right.



3 Verify what kind of tape drive you have to see if it can read QIC-24 tapes by typing:

```
mt -f <tape_device_name> status
```

QIC-150 (Archive Viper), Emulex MT-02, and Sysgen SC4000 tape drives can read QIC-24 tapes. The 4-track QIC-11 tape drives cannot read 9-track QIC-24 tapes.

4 If your tape drive cannot read QIC-24 tapes, stop now and find another tape drive that can.

Step 2. Install the Software Analyzer

1 Log in as root, and change directory to /.

2 If the *tape drive is on your local system*, extract the swa_install script from the tape by typing:

```
umask 0
tar xvfbp <device> 20 swa_install
```

3 If you are installing the software from a *tape drive on a remote system*, extract the swa_install script from the tape by typing:

```
rsh -n <hostname> "dd if=<device> bs=20b" |
   tar xvfbp - 20 swa_install
```

<device> is the name of your tape device.
<hostname> is the name of the system with the tape drive.

4 Install the software by typing "./swa_install".

The swa_install script will request the following:

- Device name of your tape device

  Enter the same <device> as above.
- Operating system for which you are installing the software

  Select the operating system used by the workstation which will run the Software Analyzer.
- Directory where you want the software to be installed

After you have answered the questions, swa_install will install the appropriate software on your hard drive in the subdirectory usr/hpcol/, then run the configure script to perform the setup required to run the Software Analyzer. If configure succeeds, *and* if you will operate the Software Analyzer only on the workstation where it was installed, skip to Step 4, "Install a SunOS patch." If configure fails, fix the indicated problem and go to Step 3, "Run the configure script."

Step 3. Run the configure script

If the swa_install script succeeded in Step 2, skip to Step 4.

- **To run the configure script, change to the subdirectory usr/hpcol/sunos_4 and run the configure script by typing:**

```
cd usr/hpcol/sunos_4
./configure
```

You should run the standalone configure script on the workstation where you originally installed the Software Analyzer, *and* on each workstation that NFS mounts the usr/hpcol/ install tree (and has a local file system). You may run the configure script as root at any time to verify that all files required by the Software Analyzer are linked to the correct locations on the local file system.

The configure script does the following:

- Installs the XKeysymDB file if the required OSF key mappings do not exist in $OPENWINHOME/lib/XKeysymDB.
- Installs two logic analyzer fonts in the $OPENWINHOME/lib/fonts directory.

$OPENWINHOME is /usr/openwin by default.

Example

Assume the HP B3740A Software Analyzer was installed on the file server apple_tree in the directory /home/apple_tree.

There are seven workstations that want to share the apple_tree HP B3740A install directory. To share this directory, you must do the following for each workstation:

1 Log in to the workstation system as root.

2 Mount the apple_tree HP B3740A install directory by typing:

```
mount apple_tree:/home/apple_tree/usr/hpcol /usr/hpcol
```

3 Change to the directory /usr/hpcol/sunos_4, and run the configure script to link the necessary HP B3740A files onto your local file system. The configure script is verbose and will tell you which files failed to link. Follow the instructions given by the configure script, and continue to run it until it succeeds.

```
cd /usr/hpcol/sunos_4
./configure
```

Step 4. Install a SunOS patch

If you have SunOS version 4.1.3_U1, you *do not* have to install this patch. The HP B3740A Software Analyzer redistributes a SunOS patch that you must install before you can use the Software Analyzer. Patch 100257-05, the "Jumbo ld patch", fixes dynamic loader defects.

This patch is located in the usr/hpcol/sunos_4/patch_dir/<patch_number> directory.

1 Check with your system administrator to see if the SunOS patch has been installed. If it has, skip to Step 5, "Set the path environment variable."

2 If the patch has not been installed, type the following commands to change to the appropriate patch directory and view the README file:

```
cd <dir>/usr/hpcol/sunos_4/patch_dir/<patch_number>
cat README
```

<dir> is the directory you installed the software analyzer in.

3 Carefully follow the directions in the README file.

CAUTION

When installing the 100257-05 patch (specifically when installing the new /usr/lib/ld.so library) follow the instructions in the README file precisely. Make sure to log in as root, and check your SunOS revision by typing "uname -r" to ensure that the correct version of ld.so is being installed.

Install the Patch on Each Workstation

You must install this patch on each Sun workstation that will run the Software Analyzer.

Step 5. Set the environment variables

1 **Set the PATH environment variable to include the <dir>/usr/hpcol/sunos_4/bin directory, where <dir> is the directory you installed the software analyzer in.**

If you are using "csh", the syntax is:

```
setenv PATH ${PATH}:<dir>/usr/hpcol/sunos_4/bin
```

If you are using "sh" or "ksh", the syntax is:

```
PATH=${PATH}:<dir>/usr/hpcol/sunos_4/bin
export PATH
```

2 **Set the HPCOL_PRODUCTS environment variable to <dir>/usr/hpcol/sunos_4, where <dir> is the directory you installed the software analyzer in.**

```
setenv HPCOL_PRODUCTS  <dir>/usr/hpcol/sunos_4
```

After you have set the environment variables, you can type "swa &" to start the Software Analyzer in background mode.

Installing the Software Analyzer (Solaris)

System Requirements

The Software Analyzer software is designed to run on a Sun SPARCsystem with SunOS version 5.3 (Solaris 2.3) or greater. The tape uses the QIC-24 data format.

In addition, your system must have:

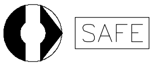
- 8 Mbytes of free disk space
- OpenWindows version 3.0 or greater
- OpenWindows non-essential MIT core clients (package SUNWowMIT)
- OpenWindows optional fonts (package SUNWowoft)
- SunSoft OSF/Motif version 1.2.2 or greater

To install the Software Analyzer on Sun workstations running the Solaris operating system, you will take the following steps:

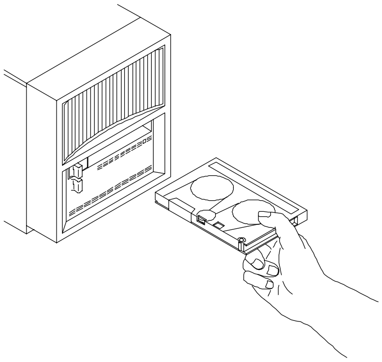
- Step 1. Load the tape.
- Step 2. Install the Software Analyzer.
- Step 3. Run the configure script (if necessary).
- Step 4. Set the environment variables.

Step 1. Load the tape

1 Verify the HP B3740A tape write-protect screw points to SAFE.



2 Insert the tape in the tape drive, and slide the lever on the tape drive to the right.



3 Verify what kind of tape drive you have to see if it can read QIC-24 tapes by typing:

```
mt -f <tape_device_name> status
```

QIC-150 (Archive Viper), Emulex MT-02, and Sysgen SC4000 tape drives can read QIC-24 tapes. The 4-track QIC-11 tape drives cannot read 9-track QIC-24 tapes.

4 If your tape drive cannot read QIC-24 tapes, stop now and find another tape drive that can.

Step 2. Install the Software Analyzer

1 Log in as root and change directory to /.

2 If the *tape drive is on your local system*, extract the swa_install script from the tape by typing:

```
umask 0
tar xvfbp <device> 20 swa_install
```

3 If you are installing the software from a *tape drive on a remote system*, extract the swa_install script from the tape by typing:

```
rsh -n <hostname> "dd if=<device> bs=20b" |
   tar xvfbp - 20 swa_install
```

<device> is the name of your tape device.
<hostname> is the name of the system with the tape drive.

4 Install the software by typing "./swa_install".

The swa_install script will request the following:

- Device name of your tape device

  Enter the same <device> as above.
- Operating system for which you are installing the software

  Select the operating system used by the workstation which will run the Software Analyzer.
- Directory where you want the software to be installed

After you have answered the questions, swa_install will install the appropriate software on your hard drive, then run the configure script to perform the setup required to run the Software Analyzer. The software will be installed under the subdirectory opt/hpcol/ in the directory you specified.

If configure succeeds, *and* if you will operate the Software Analyzer only on the workstation where it was installed, skip to Step 4, Set the PATH environment variable. If configure fails, fix the indicated problem and go to Step 3, Run the configure script.

Step 3. Run the configure script

If the swa_install script succeeded in Step 2, skip to Step 4.

- **To run the configure script, change to the directory named opt/hpcol/solar_2 by typing:**

```
cd opt/hpcol/solar_2
./configure
```

You should run the standalone configure script on the workstation where you originally installed the Software Analyzer, *and* on each workstation that has a local file system and that NFS mounts the opt/hpcol install tree. You may run the configure script as root at any time to verify that all files required by the Software Analyzer are linked to the correct locations on the local file system.

The configure script does the following:

- Checks for the existence of the SunSoft OSF/Motif 1.2.2 or 1.2.3 product under $MOTIFHOME. If $MOTIFHOME has not been defined, the script looks under /opt/SUNWmotif (for Motif 1.2.2) or under /usr/dt (for Motif 1.2.3).
- Installs two logic analyzer fonts in the $OPENWINHOME/lib/fonts directory.

$OPENWINHOME is /usr/openwin by default.

- For SunSoft OSF/Motif 1.2.3 only, creates symbolic links between versions 2 and 3 of the Motif shared libraries.

Example

Assume the HP B3740A Software Analyzer was installed on the file server apple_tree in the directory /home/apple_tree.

There are seven workstations who want to share the apple_tree HP B3740 install directory. To share this directory, you must do the following for each system workstation:

1 Log in to the workstation system as root.

2 Mount the apple_tree HP B3740 install directory by typing:

```
mount apple_tree:/home/apple_tree/opt/hpcol /opt/hpcol
```

3 Change to the directory /opt/hpcol/solar_2, and run the configure script to link the necessary HP B3740A files onto your local file system by typing:

```
cd /opt/hpcol/solar_2
./configure
```

The configure script is verbose and will tell you which files failed to link.

4 Follow the instructions given by the configure script, and continue to run it until it succeeds.

Step 4. Set the environment variables

1 Set the PATH environment variable to include the <dir>/opt/hpcol/solar_2/bin directory, where <dir> is the directory you installed the software analyzer in.

If you are using "csh", the syntax is:

```
setenv PATH ${PATH}:<dir>/opt/hpcol/solar_2/bin
```

If you are using "sh" or "ksh", the syntax is:

```
PATH=${PATH}:<dir>/opt/hpcol/solar_2/bin
export PATH
```

2 Set the HPCOL_PRODUCTS environment variable to <dir>/opt/hpcol/solar_2, where <dir> is the directory you installed the software analyzer in.

```
setenv HPCOL_PRODUCTS  <dir>/opt/hpcol/solar_2
```

After you have set the environment variables, you can type "swa &" to start the Software Analyzer in background mode.

Installing the HP Logic Analyzer Software

The HP E2450A Symbol Utility software (OPT_032, OPTC_032, or OPT5_032), the HP logic analyzer system software, and the HP module (HP 16500B only) software major version numbers must match. The HP B3740A Software Analyzer major version number does not need to match the major version number of the other parts. See the *example* software version numbers in table 5-1.

Table 5-1 **Logic Analyzer Products and Software Version Numbers Example**

Product	Software Version
HP B3740A Software Analyzer	1.00
HP 16500B System Software (SYSTEM)	3.06
HP 16555A Module Software (SYS_034)	3.06
HP E2450A Symbol Utility Software for HP 16500B (OPT_032)	3.05

* The Symbol Utility Software major software version number must match the HP logic analyzer major software version number (for example, 3.xx) for the 16500B and 1.xx for the 16500C and 1660/70 series.

The E2450A Symbol Utility Software ships with the benchtop analyzers, as well as with the Software Analyzer.

- **If you need to update the HP 16500 system and module software, copy the files from the flexible disks to the /SYSTEM subdirectory on the logic analyzer's hard disk. Then, reboot the HP 16500.**

- **If you are updating the HP 1660 or HP 1670 operating system, copy the update from the flexible disks to the flash ROMs. Updates can be ordered from your HP Sales Office.**

a Go to the System Utilities menu.

b Select the button labeled **`Update FLASH ROM`**.

c Insert the first flexible disk and follow the onscreen instructions.

The flexible disks now serve as backup copies for the files on the hard disk or in flash ROM. For more information about copying a file or updating the logic analyzer's system software, see the appropriate logic analyzer User's Guide.

Installing the Symbol Utility Software

The HP E2450A Symbol Utility Software, the HP logic analyzer system software, and the HP module software major version numbers must match.

The first time you load the symbol utility, the logic analyzer must read it from the flexible disk at powerup. To install the Symbol Utility Software on the HP logic analyzer, take the following steps:

1 **With the instrument turned off, insert the flexible disk labeled "HP E2450A Symbol Utility software" into the flexible disk drive on the logic analyzer.**

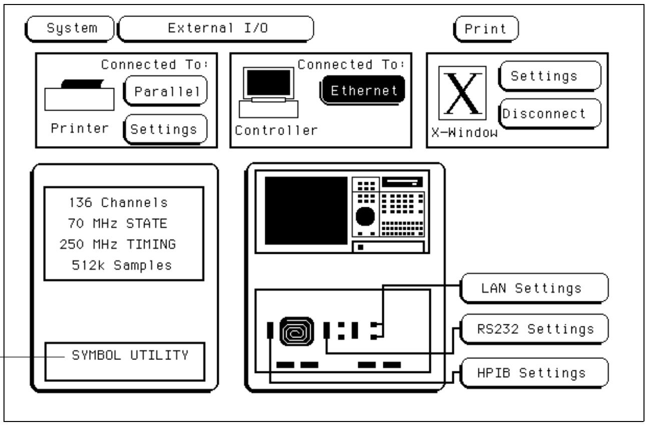
2 **Turn on the instrument.**

The software loads automatically if you inserted the flexible disk prior to turning on the instrument. The software takes several minutes to load.

CAUTION

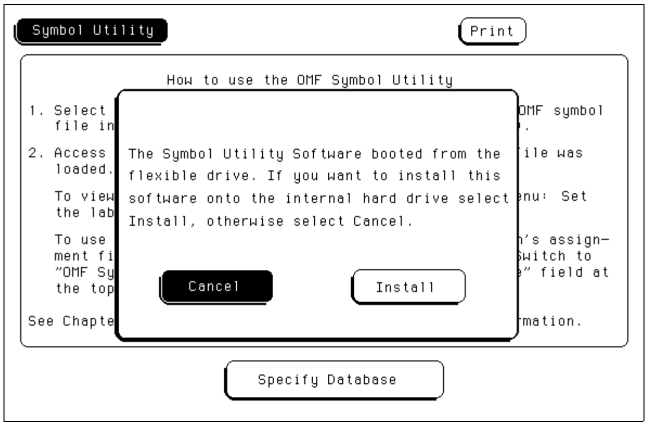
Proceed only when the light on the front of the flexible disk drive stops blinking. If you proceed before the light goes out, the installation will be incomplete and you will have to start the process over at step 1.

In the System Configuration menu (System External I/O menu on benchtop logic analyzers) menu, the Symbol Utility should now appear in the Options display. If it does not appear, repeat steps 1 and 2.



When the Symbol Utiltity is loaded on a benchtop analyzer, it shows up here.

3 Select **`System`** in the upper-left corner of the display. Then, choose **`Symbol Utility`** from the pop-up menu.



The Symbol Utility Software booted from the flexible drive. If you want to install this software onto the internal hard drive select Install, otherwise select Cancel.

This figure shows the HP 16500B logic analyzer menu. The benchtop logic analyzers are similar.

If you are using a benchtop logic analyzer, the Install/Cancel menu does not appear. The **`Install`** button appears to the right of the **`Specify Database`** button.

4 Select **`Install`** to copy the files onto the hard disk drive, or select **`Cancel`** to cancel the operation.

Installation requires about 5 minutes to complete. After the installation is complete, you can remove the Symbol Utility disk from the flexible disk drive and store it as a backup. The Symbol Utility will load automatically whenever you turn on the logic analyzer.

CAUTION

Proceed only when the light on the front of the flexible disk drive stops blinking. If you proceed before the light goes out, the installation will be incomplete and you will have to start the process over at step 1.

AOUT This output format is used by the GNU tool chain for various processors, including the 68000 family. Other varieties of AOUT are not supported.

benchtop logic analyzer A small, standalone HP logic analyzer that is not part of the HP 16500 logic analysis system. As of this edition, only the HP 1660C/CS-series and the HP 1670-series can run the Software Analyzer.

browser A pop-up menu which lets you examine and select OMF symbols.

data label A symbol that refers to an arbitrary value, much like the labels that are available in the logic analyzer without the Symbol Utility. All other symbols refer to addresses. Data labels can be defined only in the General-Purpose ASCII symbol file format.

database file The file created by the symbol utility or nsbuild. The file is organized so that the Symbol Utility can efficiently look up the symbol associated with any address or label value. Database files have a ".ns" file name extension.

ELF/DWARF This OMF is a portable format consisting of the ELF (Executable and Linkable Format) and the DWARF (Debugging Information Format) for various processors, including Intel 80960, PowerPC, and MIPS.

ELF/STABS This OMF is a portable format consisting of the ELF (Executable and Link Format) and the "stabs" debug format. The "stabs" format is derived from "The Stabs Debug Format," revision 2.125, produced by Cygnus Support. Deviations from this specification are not supported.

GPA The General-Purpose ASCII file format. These symbol files are created by the user with an ASCII text editor.

HP-MRI IEEE-695 The OMF format produced by HP and MRI cross-compilers and cross-assemblers. This format is fully supported by the Symbol Utility IEEE-695 reader.

IEEE-695 A particular format for an OMF. This format is usually produced by language tools for non-Intel microprocessors. Some language tools that claim to output this format may deviate from the IEEE-695 standard in ways that make the OMF file unreadable by the Symbol Utility reader.

language tool chain A set of programs used to generate executable code. For example, a compiler, assembler, and linker.

loaded file A database file that has been loaded into the Symbol Utility.

measurement module Any of the cards that can be inserted into the slots in the HP logic analyzer, such as an oscilloscope, pattern generator, or state analyzer. The HP E2450A Symbol Utility requires a state analyzer measurement module.

nsbuild A utility that allows you to build symbol database files on your host computer rather than on your logic analyzer. The nsbuild method is faster than the logic analyzer method.

OMF A file that contains mixed object code and symbol information. OMF stands for Object Module File. The OMF is produced by the language tool chain. These files are also referred to as symbol files.

OMF286 A particular format for an OMF. This format is usually produced by language tools for Intel 80x86-series and Pentium microprocessors running in real or protected mode.

OMF386 A particular format for an OMF. This format is usually produced by language tools for Intel 80x86-series or Pentium microprocessors running in real or protected mode.

OMF86 A particular format for an OMF. This format is usually produced by language tools for Intel 80x86-series and Pentium microprocessors running in real mode only.

OMF96 A particular format for an OMF. This format is usually produced by language tools for the Intel 80196 family of microprocessors.

point symbol A symbol that refers to a single address.

procedure A term used in the Symbol Utility to refer to procedures, functions, subroutines, or their equivalents.

range symbol A symbol that refers to a range of addresses, such as a module or procedure.

reader A program that generates a database file from an OMF file on the logic analyzer.

symbol database file See database file.

symbol file A file that contains symbol debugging information. This file is produced by the language tool chain. These files are also referred to as OMF files.

TI-COFF An OMF produced by language tools for Texas Instruments microprocessors.

user symbol table A table containing the symbols that are created by the user, using the Symbols option in the Format menu. This is the standard symbol capability of the HP logic analyzer.

Win32s A Microsoft product that allows 32-bit Windows applications to run in MS Windows 3.1.

Index

Document Warranty

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

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety

This apparatus has been designed and tested in accordance with IEC Publication 348, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

Warning

- Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.
- Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock of fire hazard.
- Service instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.
- If you energize this instrument by an auto transformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.
- Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.
- Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.
- Do not install substitute parts or perform any unauthorized modification to the instrument.
- Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.
- Use caution when exposing or handling the CRT. Handling or replacing the CRT shall be done only by qualified maintenance personnel.

Safety Symbols

⚠

Instruction manual symbol: the product is marked with this symbol when it is necessary for you to refer to the instruction manual in order to protect against damage to the product.

⚡

Hazardous voltage symbol.

⏚

Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.

Hewlett-Packard
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are the buyer's sole and exclusive remedies. Hewlett-Packard shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales Office.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard 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.

About this edition

This is the sixth edition of the *HP B3740A Software Analyzer User's Guide.*

Printed in USA.

Edition dates are as follows:

First edition, May 1994

Second edition, June 1994

Third edition, July 1994,
B3740-97002

Fourth edition, March 1995,
B3740-97003

Fifth edition, November 1995,
B3740-97004

Sixth edition, February 1997,
B3740-97005

Many product updates do not require manual changes; and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.