# HP 83224A IBASIC Developer's Tool Kit

## *User's Guide*

Software Version A.00.02 and above

**HP Part No.  83224-90002**
**Printed in U. S. A.**
**March 1997**

**Rev. B**

# In This Book

This book will teach you how to install and use the HP 83224A IBASIC Developer’s Tool Kit. It will also introduce you to some common tasks performed by users of this software. Throughout this manual, The HP 83224A will be referred to as "IBASIC Tool Kit."

## How to Use This Manual

This manual has six chapters directly relevant to the use of the IBASIC Tool Kit: General Information; Installation; Getting Started; Operational Overview; Basics of Code, Library and Procedure Files; and Reference Information.

1. The "General Information" chapter contains general information about the HP 83224A IBASIC Developer’s Tool Kit for Windows.
2. The "Installation" chapter helps you install the IBASIC Tool Kit.
3. The "Getting Started" chapter provides you with information “at a glance” to get you started using the Software Development Program (DEV_PC).
4. The "Operational Overview of DEV_PC" provides you with an overview of DEV_PC and its primary purposes of use. You are also given an overview of the DEV_PC “Configure System” and its system parameters.
5. The "Basics of Code, Library, and Procedure Files" provides basic instructions on how to use DEV_PC for developing code and for modifying and creating library and procedure files.
6. The "Reference Information" chapter provides reference information about: the files found on the DEV_PC floppy disk, the library and procedure file architecture, DEV_PC functional descriptions, a sample IBASIC program with code comments.

# Contents

# 1

# General Information

This chapter contains general information about the HP 83224A IBASIC Developer’s Tool Kit for Windows™ [1] (hereafter referred to as the “IBASIC Tool Kit”). Prior to installing the IBASIC Tool Kit, please read the information in the following pages to ensure your success with this product.

[1] Windows is a U.S. trademark of Microsoft® Corporation.

# HP 83224A IBASIC Developer’s Tool Kit for Windows

## IBASIC Tool Kit Contents

The HP 83224A IBASIC Developer’s Tool Kit for Windows has the following contents, depending upon the option you ordered:

**Option 001:**

- HP BASIC for Windows (HP E2060B)
- HP-IB Interface Card (HP 82335B)
- HP Test Set Program-Development Software (DEV_PC)
- HP-IB Cable (HP 10833B)
- Documentation for the IBASIC Tool Kit

**Option 002:**

- HP Test Set Program-Development Software (DEV_PC)
- Documentation for the IBASIC Tool Kit

***NOTE:*** If you are missing any part of the IBASIC Tool Kit, contact your Hewlett-Packard office where you placed the order; notice that option 002 does not have the same contents as those found in option 001.

## Before Using the IBASIC Developer’s Tool Kit

Before you start using the IBASIC Tool Kit, the next chapter of this manual instructs you to do the following:

1. Install the HP-IB Interface Card*
2. Configure the PC*
3. Install HP BASIC for Windows*
4. Install the DEV_PC Software
5. Run the DEV_PC Software
6. Verify installation of the IBASIC Tool Kit

***NOTE:*** * If you purchased HP 83224A Option 002, you only need to perform steps 4 through 6 as outlined in the installation section of this manual since it would be assumed you already have an HP-IB card and the HP BASIC for Windows software installed in your PC.

## System Overview

The Software Development program DEV_PC, part of the IBASIC Tool Kit, serves three primary purposes:

1. The DEV_PC software provides you with a user-interface to facilitate the creation or modification of library and procedure files that are used by the Tests Subsystem within the Test Set.
2. DEV_PC runs within HP BASIC for Windows to serve as a programming environment from where you can develop and run code which conforms to the IBASIC standards and syntax specific to the HP 8920 and HP 8921 family of Test Sets. DEV_PC provides function keys to better facilitate the writing and transfer of code between the Test Set and PC than what is done by using HP BASIC for Windows by itself.
3. DEV_PC allows you to convert LIF formatted code, library, and procedure files from the HP 8920 and HP 8921 Test Sets into the DOS format for use on the HP 8920B.

These three purposes are covered in depth later on in this manual.

The components of IBASIC Tool Kit are shown in figure 1 on page 10:

❒ The HP-IB interface card and HP-IB cable provide the hardware connection for transferring code, library, and procedure files between the PC and Test Set.

❒ The HP BASIC for Windows software provides a shell to run DEV_PC and is the interface used to support the HP-IB card driver for controlling the Test Set. The ID Module must be installed on the PC for the HP BASIC software to operate.

❒ The DEV_PC software runs in HP BASIC for Windows and is the primary tool you’ll use for IBASIC program development.

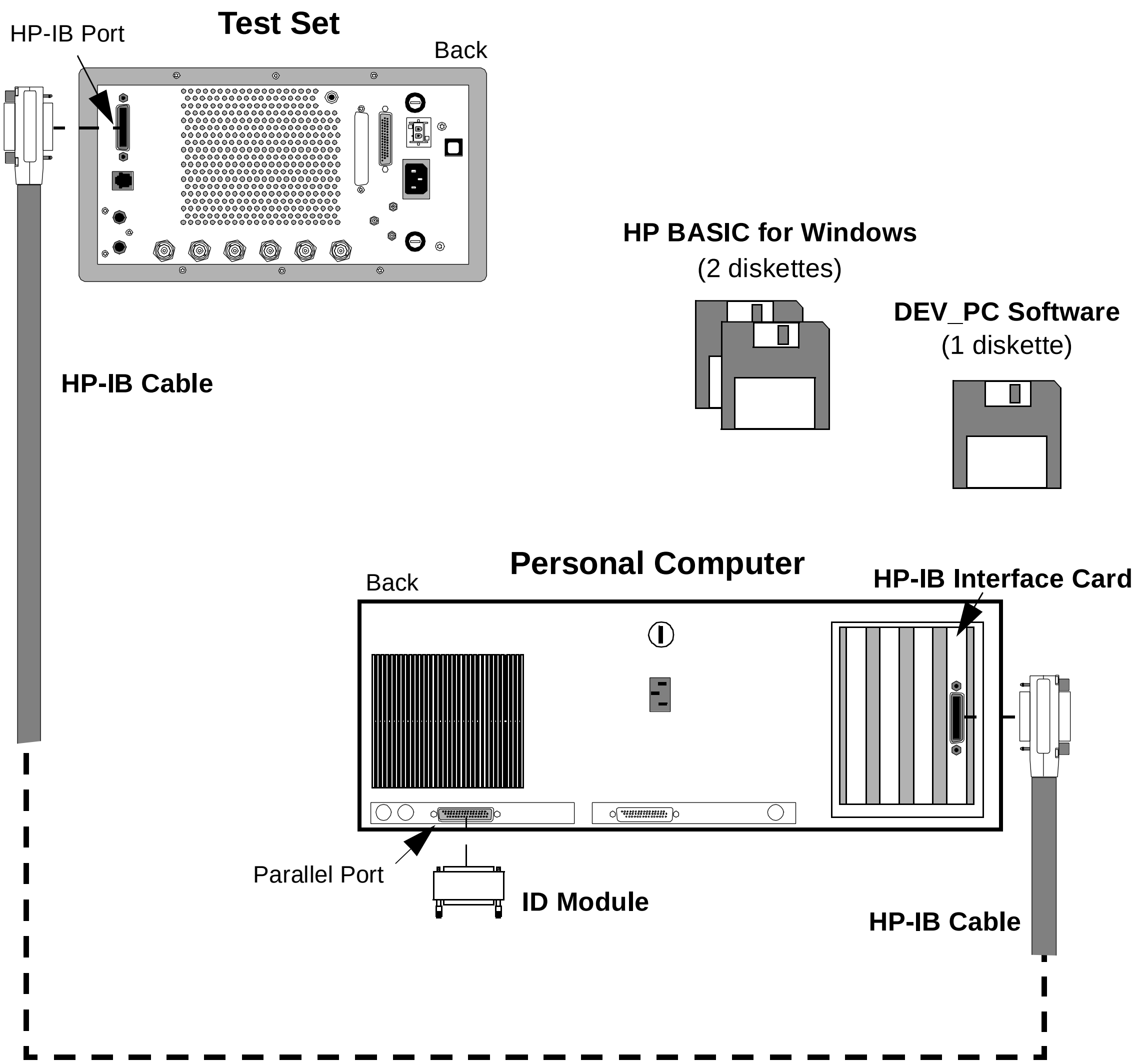


**Figure 1** **IBASIC Tool Kit Installation**

## Needing Help?

In addition to the manuals that come with the HP-IB Interface Card and HP BASIC for Windows, we recommend you have the following manuals:

❒ *HP Instrument BASIC Programmer's Guide*
❒ *HP 8920A Programmer's Guide*
❒ *HP 8920B Programmer's Guide*
❒ *HP 8921 Programmer's Guide*

***NOTE:*** The HP 8920 and HP 8921 Test Sets all use IBASIC Revision 1.8, and all HP 8920B Test Sets use IBASIC Revision 2.0. The difference in revisions is that the command-language set of IBASIC Revision 2.0 has more commands than Revision 1.8.

If you still need assistance, you can get help from the following locations:

- **For the HP-IB Interface Card:** Complimentary start-up assistance with the HP-IB interface card is available for a limited time starting from the customer's first call. Refer to the installation manual that comes with this product for installation instructions and for help if it is needed.
- **For HP BASIC for Windows:** Complimentary start-up assistance with HP BASIC for Windows is available for a limited time starting from the customer's first call. Refer to the installation manual that comes with this product for installation instructions and for help if it is needed.
- **For the DEV_PC software:** Complimentary start-up assistance for the DEV_PC software is available for 90 days starting from the customer's first call. Depending on your question, there are a few means by which you can get help:
  - We encourage you to first try and find an answer in this manual!
  - Contact your local Hewlett-Packard sales office if you need consulting assistance with IBASIC programming. Hewlett-Packard system engineers are available for hire.
  - For support in the United States or in Canada only, call the Factory Application Assistance HOT-line number at 800-922-8920 if you have specific questions about using the DEV_PC software. Normal hours for this service are from 8 am to 5 pm Pacific Time.
  - Fax your questions to 509-921-3700, attention IBASIC Tool Kit Assistance.
  - Leave an E-mail message at **`online_support@spk.hp.com`** and be sure to include your daytime phone number.

***NOTE:*** Hewlett-Packard phone support does not include tutorial training for those needing help learning how to program. Contact your local Hewlett-Packard sales office for system engineer consulting assistance; this service is provided on a fee basis.

## Equipment Needed

The IBASIC Tool Kit requires the following equipment:

### Personal Computer

An IBM compatible 386 CPU- PC or better with at least 4 MB RAM memory (8 MB recommended), MSDOS®[1] 5.0, and Windows 3.1 or Windows 95.

***NOTE:*** The IBASIC Tool Kit is not compatible with Windows NT.

### Test Sets Supported

The HP 8920 and HP 8921 family of test sets are supported by the IBASIC Tool Kit regardless of the firmware version in the test set.

### Memory Cards

The test set you use determines the type of memory card you must have. Both EPSON and PCMCIA SRAM memory cards use a battery which should be replaced after a year of use. A detailed description of memory cards is found in the *HP 8920 and HP 8921 Programmer’s Guide.*

The HP 8920A/D and HP 8921 test sets require use of an EPSON-type SRAM memory card to store code, procedure, and library files. Some typically used EPSON SRAM memory cards are listed in table 1.

**Table 1** **EPSON SRAM Memory Cards**

| Memory Size * | Hewlett-Packard Part Number |
|---|---|
| 32 KB | HP 85700A |
| 128 KB | HP 85702A |
| 256 KB | HP 85704A |
| 512 KB | HP 85705A |
| * All require battery HP part number 1420-0383, or generic part number CR2016 | |

The HP 8920B test set requires use of a PCMCIA memory card to store code, procedure, and library files. Some typically used PCMCIA SRAM memory cards are listed in table 2.

[1] MSDOS is a U.S. trademark of Microsoft® Corporation.

**Table 2** **PCMCIA SRAM Memory Cards**

| Memory Size * | Hewlett-Packard Part Number |
|---|---|
| 64 KB | HP 83230A |
| 1 MB | HP 83231A |
| * All require battery HP part number 1420-0509, or generic part number CR2025 | |

NL

Batterij niet weggooien, maar inleveren als KCA.

# 2

# Installation

This chapter helps you install the IBASIC Tool Kit.

## Procedure for Installing the IBASIC Tool Kit

To install, follow the procedure steps 1 through 5. If you are installing the HP 83244A Option 002, start at step 4. The final step of this procedure, step 6, verifies that the IBASIC Tool Kit has been correctly installed; during this step, you will create and run a simple program, and then download the program into a Test Set.

### 1. Install the HP-IB Interface Card

To complete this step, please follow instructions found in the *Installing the HP-IB Interface* booklet that comes with the HP-IB card. In this booklet, you are directed to unpack and set the configuration switches on the HP-IB card, install the HP-IB card in the PC, and connect the HP-IB cable from the PC to the Test Set.

*NOTE:* It is not necessary when installing the IBASIC Tool Kit to load the “Interactive HP-IB Environment” software which comes with the HP-IB Interface Card. Instead, by using HP BASIC for Windows with the DEV_PC software you are provided with a better interface for interactive control of the Test Set over HP-IB.

*NOTE:* In the **`I/O CONFIGURE`** menu on the Test Set, make sure that the **`Mode`** function is set to **`Talk&Lstn`**, otherwise the PC will display an HP-IB “error 167” message when HP BASIC for Windows is installed during step *3.Install HP BASIC for Windows.*

### 2. Configure the PC

In this step, you will modify the CONFIG.SYS and AUTOEXEC.BAT files. These modifications are NOT done automatically when you install HP BASIC for Windows or the DEV_PC software!

**Configure your PC’s Memory Manager.**

If your PC uses an expanded memory manager, you must configure it to not use the HP-IB interface address space by modifying the **`CONFIG.SYS file`**. A thorough description of the memory manager’s configuration settings is found in the *Installing the HP-IB Interface* booklet.

For example, if your PC is configured to use the **`HPEMMGR.SYS`** or **`EMM386.SYS`** drivers, you would modify the **`config.sys`** file at the driver statement to include the memory-manager configuration **`X=DC00-DFFF`**.

### Configure your PC for File Sharing

In order to allow file-sharing and locking capabilities you'll need to add the share utility to the **`AUTOEXEC.BAT`** file. This utility is especially useful to prevent a newer file from writing over an older one if more than one copy of the file is accessed at the same time.

1. From Windows, select the **`MS-DOS Prompt`** and change directory (cd..) to the root level prompt **`C:\>`** .
2. Type **`edit autoexec.bat`**, then press the Enter key and type **`share.exe`** after the PATH statement for DOS.
3. Exit the on-line text editor saving this change to the **`autoexec.bat`** file.

***NOTE:*** In order for changes to the CONFIG.SYS and AUTOEXEC.BAT files to be in effect, reboot your computer.

### 3. Install HP BASIC for Windows

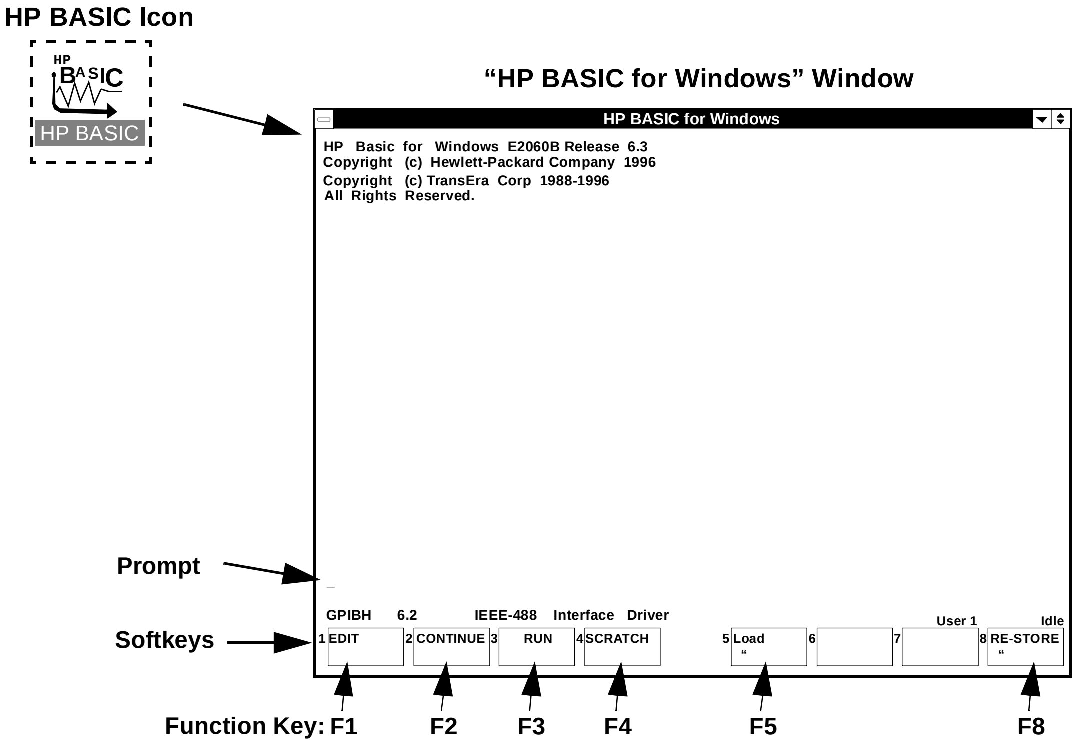
To complete this step, please follow instructions found in the *Installing and Using Guide* that comes with HP BASIC for Windows. When the setup is complete, there will be a program group titled “HP BASIC for Windows” from where you select the **HP BASIC** icon (see figure 2, "HP BASIC Icon and Window").

***NOTE:*** The IBASIC Tool Kit will not work unless the ID module is installed on the PC’s parallel port. If you select the HP IBASIC icon without the ID module connected to the PC, you will observe an error message prompting you to connect one.

**Configure HP BASIC with HP-IB Card Driver**

To use HP BASIC for Windows with the HP 82335 HP-IB Interface Card, you must change a line of code in the HP BASIC AUTOST program in order to load the GPIBH driver. Refer to the *Installing and Using Guide* for HP BASIC for Windows if you need more information.

1. From Windows, double-click on the **HP BASIC** icon. The HP BASIC for Windows screen appears, see figure 2 .



```
HP  Basic  for  Windows  E2060B Release  6.3
Copyright  (c)  Hewlett-Packard Company  1996
Copyright  (c) TransEra  Corp  1988-1996
All  Rights  Reserved.

_
GPIBH    6.2        IEEE-488   Interface   Driver
```

**Figure 2** **HP BASIC Icon and Window**

2. At the prompt, type **`edit 310`**, then press the Enter key and modify line 310 to be:

```
310 LOAD BIN “GPIBH;IN 3”
```

3. Save the change to line 310 by selecting the 8 softkey and typing:

```
RE-STORE “AUTOST”
```

*NOTE:* The HP-IB interface card driver **`gpibh.dw6`** is found in directory **`c:\hpbasic`** and is used with the HP 82335 . If you have other HP-IB interface cards such as the HP 82340A or the HP 82341B, you would use the driver **`gpib.dw6`** and change line 310 accordingly.

### Verify that HP BASIC for Windows talks to the Test Set

The following procedure verifies that the preceding steps of installing the HP BASIC for Windows has been successful.

1. With HP BASIC for Windows running, enter the following command statement:

```
output 714;”*rst”
```

*NOTE:* Step 1 assumes that your HP-IB card is set to interface select code 7 and the Test Set is set to HP-IB address 14 in the **`I/O CONFIGURE`** screen.

2. After you press the Enter keyboard key, the Test Set will perform a reset and the **`RX TEST`** screen appears on the Test Set’s display with **`RL`** in the upper-right corner of the screen, see figure 3 below. The **`R`** indicates the Test Set is in “remote” mode; **`L`** indicates that it is configured to “talk and listen”.

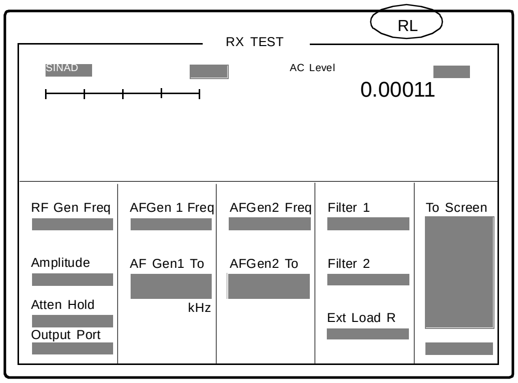


**Figure 3** **Test Set’s RX TEST Screen**

3. If you get an error message, verify that you have entered the command statement shown in step 1 above, and that all preceding steps for installing the HP-IB Interface Card and HP BASIC for Windows are done.

## 4. Install the DEV_PC Software

Do not proceed with this step unless you have verified that the HP-IB Interface Card is correctly installed and that it works with HP BASIC for Windows to talk to the Test Set!

Use the following steps to install the Software Development Program (DEV_PC):

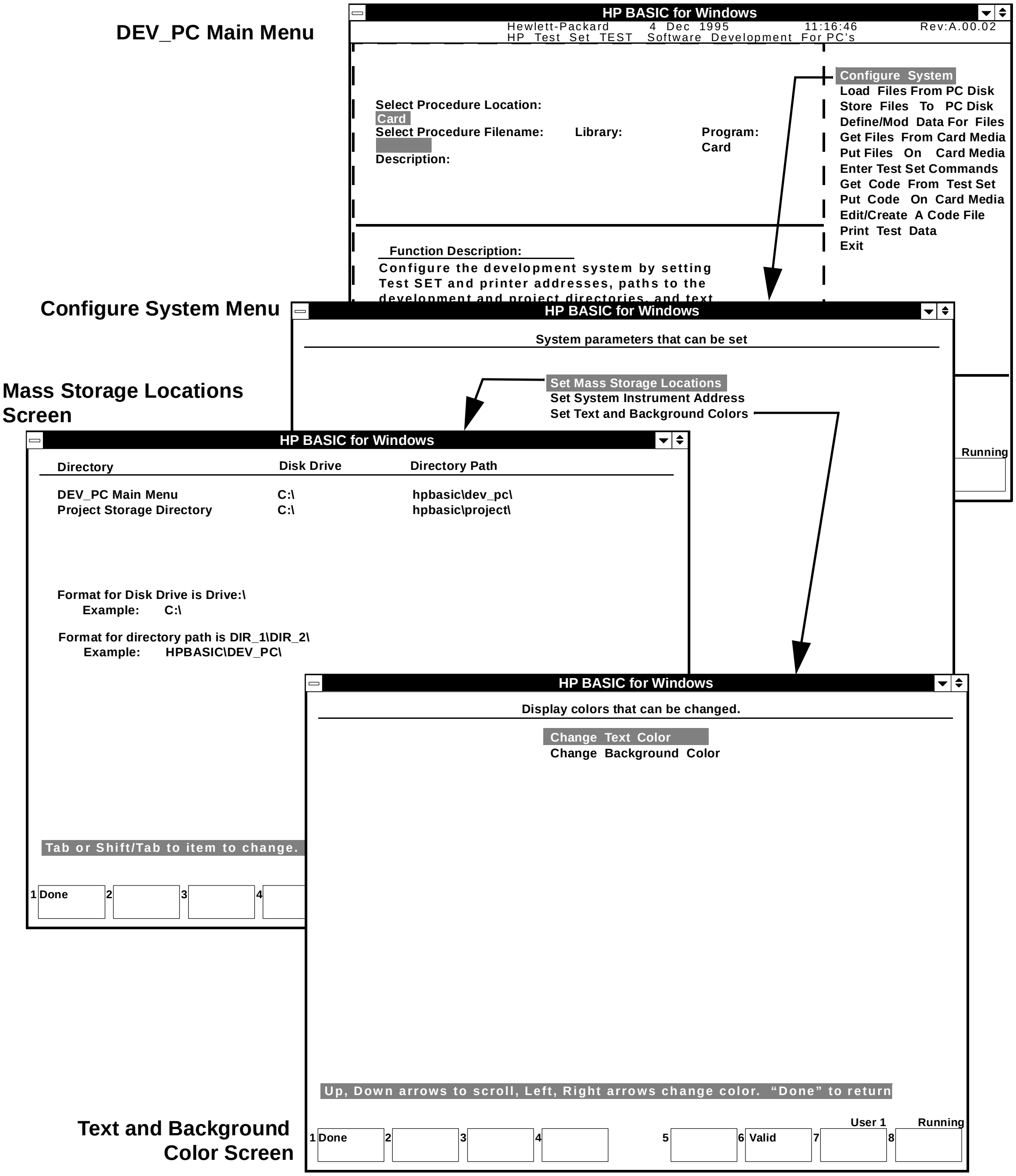
1. Make a backup copy of the DEV_PC floppy disk.
2. Make your project storage directories.

   For example, you may choose to keep your projects under a directory such as **`c:\hpbasic\project1`**; you will make the directory and then later on specify this directory path as a mass-storage location when DEV_PC is running.
3. Make a directory **`c:\hpbasic\dev_pc`** and copy all files from the DEV_PC floppy disk to this directory. (The files may be installed on a network.)

## 5. Run the DEV_PC Software

The following steps tell you how to load and run the DEV_PC software. You also are instructed to configure the mass-storage locations for the project directory you previously created in step *4.Install the DEV_PC Software*.

1. From Windows, double-click on the **`HP BASIC`** icon.
2. Type **`msi “dev_pc”`** then press the Enter keyboard key.
3. Select the 5 softkey and type **`get “dev_pc”`**, then press the Enter keyboard key.
4. Select the 3 softkey to run DEV_PC. The DEV_PC main menu is displayed on your PC monitor. See figure 4, "Configure System Functions," on page 21.
5. Select **`Configure System`** from the main menu, then select **`Set Mass Storage Locations`**.
6. Use the Tab key to select the directory path area for the **`Project Storage Directory`**; then enter the directory path you made in step *4. Install the DEV_PC Software*. For example you might enter, **`hpbasic\project1\`**
7. Select the 1 softkey **`Done`** and then the 1 softkey again to store the mass-storage locations.
8. You may then want to change the default colors of the DEV_PC screen by selecting the **`Set Text and Background Colors`** function of the Configure System menu before exiting back to the main menu, see figure 4 on page 21.



Hewlett-Packard 4 Dec 1995 11:16:46 Rev:A.00.02
HP Test Set TEST Software Development For PC's

Select Procedure Location:
Card
Select Procedure Filename:
Library:
Program:
Card
Description:

Function Description:
Configure the development system by setting Test SET and printer addresses, paths to the development and project directories, and text

System parameters that can be set

| Directory | Disk Drive | Directory Path |
|---|---|---|
| DEV_PC Main Menu | C:\ | hpbasic\dev_pc\ |
| Project Storage Directory | C:\ | hpbasic\project\ |

Format for Disk Drive is Drive:\
Example: C:\

Format for directory path is DIR_1\DIR_2\
Example: HPBASIC\DEV_PC\

Tab or Shift/Tab to item to change.

Display colors that can be changed.

Up, Down arrows to scroll, Left, Right arrows change color. "Done" to return

**Figure 4** **Configure System Functions**

### 6. Verify Installation of the IBASIC Tool Kit

The following steps verify that the IBASIC Tool Kit was correctly installed. You will use DEV_PC to create and run a simple program, you then download the program to the Test Set and run it, finally you'll put the program on a memory card.

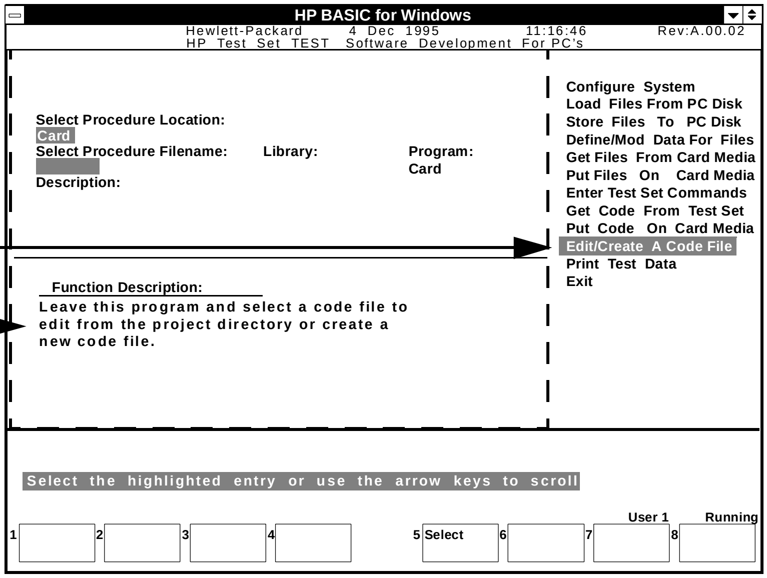
#### Create and Run a Simple Program to Set Frequency and Amplitude

**1.** Use the keyboard's up/down arrow keys to select **`Edit/Create A Code File`** function on the DEV_PC main menu (see figure 5, "Edit/Create A Code File Function"). Notice the "Function Description" describes each function as you move the selection bar.

DEV_PC Main Menu

*"Edit/Create A Code File"*

*Description*



HP BASIC for Windows

```
Hewlett-Packard        4 Dec 1995             11:16:46          Rev:A.00.02
HP Test Set TEST   Software Development For PC's
```

Select Procedure Location:
Card
Select Procedure Filename:    Library:    Program:
Card
Description:

Function Description:
Leave this program and select a code file to edit from the project directory or create a new code file.

Select the highlighted entry or use the arrow keys to scroll

User 1    Running

**Figure 5** **Edit/Create A Code File Function**

**2.** You are prompted to terminate this program and start development. Select the 1 softkey **`Yes`**.

**3.** You are prompted to type **`scratch`**, press the Enter key, then type **`edit`** and press the Enter key.

If any program files (with the **`.pgm`** file extension) exist in the project directory, they show up on the monitor for you to "get" and then edit.

4. Type in the following 11 line program.

```
10 !FREQ !Comment out a name for the program; it must be on 1st line of code.
20 Uut=714 !Test Set's address to be used by PC for control
30 OUTPUT Uut;"*rst"
40 WAIT 2
50 OUTPUT Uut;"rfg:freq 123 mhz" !sets RF generator to 123 MHz
60 OUTPUT Uut;"rfg:ampl -20 dbm" !sets RF amplitude to -20 dBm
70 OUTPUT Uut;"disp dupl" !sets up the Duplex Test screen
80 OUTPUT Uut;"meas:rfr:freq:abs?" !query value of RF generator
90 ENTER Uut:A !put value in "A"
100 PRINT "Frequency is:";A !Print value of "A" to PC or Test Set
110 END
```

5. Overwrite line number `120` when it appears with the word `run`, then press the Enter key.

   Notice the Test Set goes to the Duplex screen, and the results displayed on the PC should be: `Frequency is: 1.23E+8`

***NOTE:*** If an error occurs on the Test Set, verify that your program is identical to what is shown in the above example.

6. Edit line 20 to read `20 Uut=800` (this change prepares the code to run on the Test Set after it's downloaded. The Test Set's internal IBASIC bus has an 800 address).

### Download Simple Program to Test Set and Run It

7. Download the 11 line program into the Test Set by selecting the 1, 2, 3, and 4 softkeys in that order. (These softkeys perform the necessary downloading commands to put a resident program into the Test Set's RAM memory.)

8. On the Test Set, press the LOCAL and TESTS keys, then select the **`IBASIC Controller`** and select the K1 **`Run`** user key.

Notice what happens to the Test Set, you should see the following results displayed:

`Frequency is: 1.23E+8`

9. Use the knob to type **`LIST`**, then select **`Done`** at the top of the list of **`CHOICES`** to list all 10 lines of code on the Test Set.

10. On the PC, select the 6 softkey and type `RE-SAVE "FREQ.PGM"` and press the Enter key. (This step puts the program you wrote on the PC's hard-disk at the directory path specified for program files.)

### Put the Simple Program on a Memory Card

11. Select the 8 softkey to `Run DEV_PC`.

12. Select the **Put Code On Card Media** function then enter a filename (use the same name you put on line 10 of the code) for the program. Ensure a SRAM memory card is inserted into the Test Set.

13. Follow the on-line instructions. When finished, you should see the PC’s response **Storage successful** displayed and a catalog of the memory card’s contents shown on the Test Set.

14. Select the 1 softkey **Done** to display the main menu for DEV_PC.

## Installation Wrap Up.

Once you’ve verified that the IBASIC Tool Kit has been correctly installed, proceed to the *Getting Started* chapter and find the section that best describes your application.

# 3

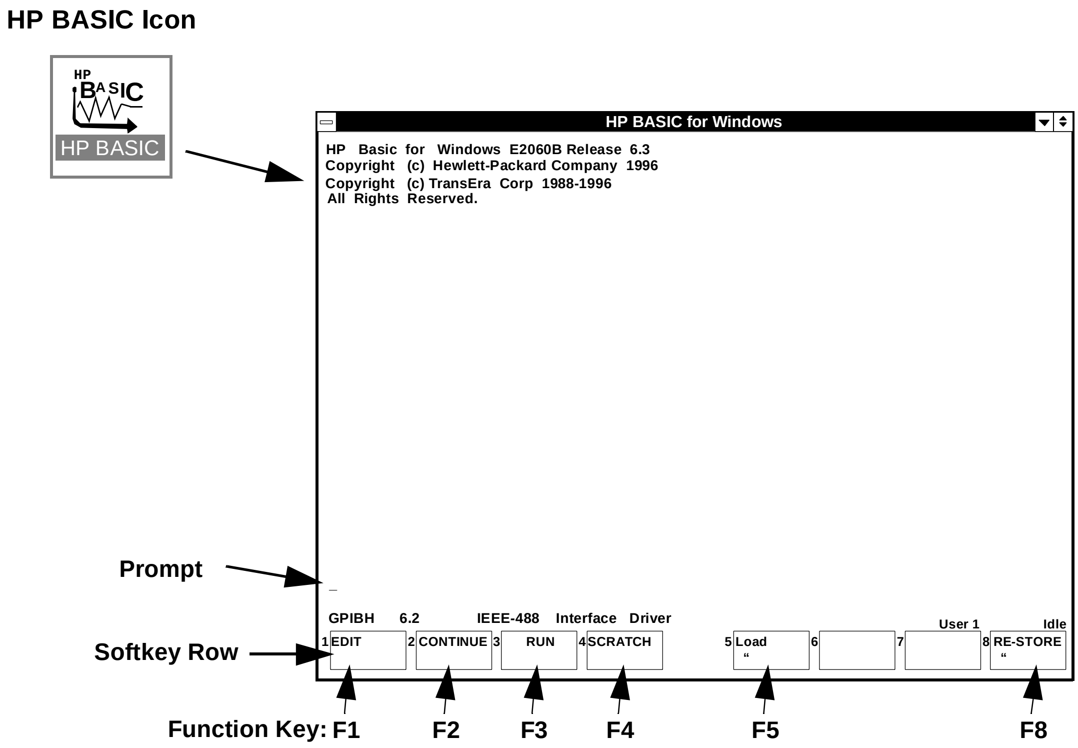
# Getting Started

This chapter provides you with information “at a glance” to help you get started using the Software Development Program (DEV_PC). The DEV_PC main menu is explained, and a helpful note is given about the DEV_PC main menu display.

# DEV_PC Main Menu at a Glance

To access the DEV_PC main menu:

1. From Windows, double-click on the **`HP BASIC`** icon. The HP BASIC for Windows window appears. Refer to figure 6 for reference.
2. At the prompt, type **`msi "dev_pc"`** then press the keyboard's Enter key.
3. Select the 5 softkey or press the F5 key and type **`get "dev_pc"`**, then press the Enter key.
4. Select the 3 softkey or press the F3 key to run the DEV_PC software. The DEV_PC main menu as shown in figure 7 on page 27 appears on your PC monitor.



```
HP  Basic  for  Windows  E2060B Release  6.3
Copyright  (c)  Hewlett-Packard Company  1996
Copyright  (c) TransEra  Corp  1988-1996
All  Rights  Reserved.

_
GPIBH     6.2          IEEE-488   Interface   Driver                    User 1          Idle
```

**Figure 6** **The "HP BASIC for Windows" Window**

The main menu has three major areas of interest (refer to figure 7 for reference):

1. **Procedure Location, Filename, and Library area.**

   This is a display area only. You cannot select any of these fields from the PC, as you can the same corresponding fields on the Test Set. This area shows information about the current test procedure, it's location, filename, and library used (if any). This area is updated whenever a test procedure is loaded from the Test Set using the main-menu function **`Get Files From Card Media`**, or when you use the main-menu function **`Define/Mod Data For Files`**.

2. **The DEV_PC function description area.**

   This area provides a brief description of each DEV_PC function that is selectable from the main menu.

3. **The DEV_PC function list area.**

   Select the main-menu DEV_PC functions from this area by using the keyboard's up or down arrow keys. As the selection bar moves up or down, the corresponding **`Function Description`** changes as well.

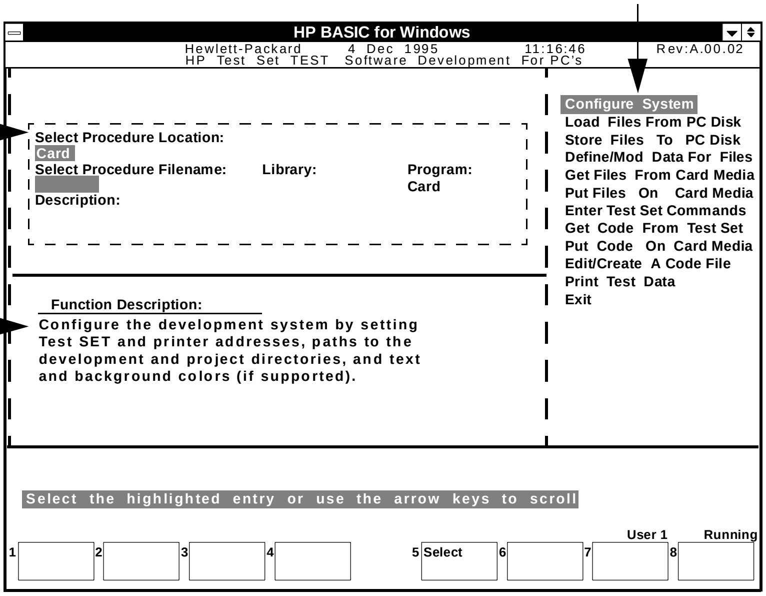
***NOTE: ON THE PC***

The mouse does not work with DEV_PC other than to select the 1 through 8 softkeys at the bottom of the DEV_PC screen.

*3. List of DEV_PC functions.*

*1. Area identifying the location and filename of the Procedure and Library files.*

*2. Description of the DEV_PC function selected, for example, this is a description of the "Configure System" function.*



HP BASIC for Windows

Hewlett-Packard 4 Dec 1995 11:16:46 Rev:A.00.02
HP Test Set TEST Software Development For PC's

Select Procedure Location:
Card
Select Procedure Filename: Library: Program:
Card
Description:

Function Description:
Configure the development system by setting Test SET and printer addresses, paths to the development and project directories, and text and background colors (if supported).

Select the highlighted entry or use the arrow keys to scroll

**Figure 7** **The DEV_PC Main Menu**

## Setting the Geometry Switch on the DEV_PC Main Menu Display

If you are accustomed to using the Window minimize, maximize, or restore functions as you switch between Windows, you may find the orientation of text on the DEV_PC main menu shifted, or even blanked. If this happens you can permanently fix the display problem by setting a command-line switch to the **`HP BASIC`** icon. Other information about this command-line switch is found in the *Installing and Using Guide* for HP BASIC for Windows.

**From the Program Manager.**

1. At the Program Group for HP BASIC for Windows, single-click on the **`HP BASIC`** icon and select **`File`** then **`Properties`**.
2. In the Command Line area, type **`HPBASIC.EXE -geometry 80x30+0+0 -gr on`** then select OK.
3. Double-click on the HP BASIC icon and re-load DEV_PC.

4

# Operational Overview of DEV_PC

This chapter provides you with an overview of DEV_PC and its three primary purposes of use. Finally, you are given an overview of the DEV_PC “Configure System” and its three system parameters that can be set.

# Operational Overview

### The Purpose of this Chapter

This chapter covers the following:

- Creating new, or modifying existing programs.
- Using the Tests Subsystem.
- Converting HP 8920 or HP 8921 code for use on the HP 8920B.
- The DEV_PC “Configure System”.

*WARNING:* **Use of the DEV_PC software to make copies of Hewlett-Packard software for purposes other than those stated in the Software License Agreement is strictly prohibited!**

*NOTE:* It’s to your advantage as you proceed with this chapter to have the following documentation available:

❒ *HP Instrument BASIC Programmer’s Guide*
❒ *HP 8920A Programmer’s Guide*
❒ *HP 8920B Programmer’s Guide*
❒ *HP 8921A Programmer’s Guide*

### Creating New or Modifying Existing Programs.

Program code is uploaded from the Test Set or newly created on the PC. The code may even be taken from an HP 11807 source. In any case, the PC is used to control the Test Set during program development. When the code is debugged, it can be downloaded into the Test Set’s RAM memory, and stored on a memory card. A backup file kept on the PC’s hard disk ensures that you always have a copy of the source code.

Figure 8, "Editing or Creating Program Code at a Glance," on page 31 shows you how to interact with the DEV_PC main menu to create new or modify existing programs.

**`Get Code From Test Set`**: Get code that is presently loaded in the Test Set’s IBASIC RAM memory. You will be queried for a file name. The file will be stored in the project directory under that name.

**`Edit/Create A Code File`**: Leave this program (DEV_PC) and select a code file to edit from the project directory or create a new code file.

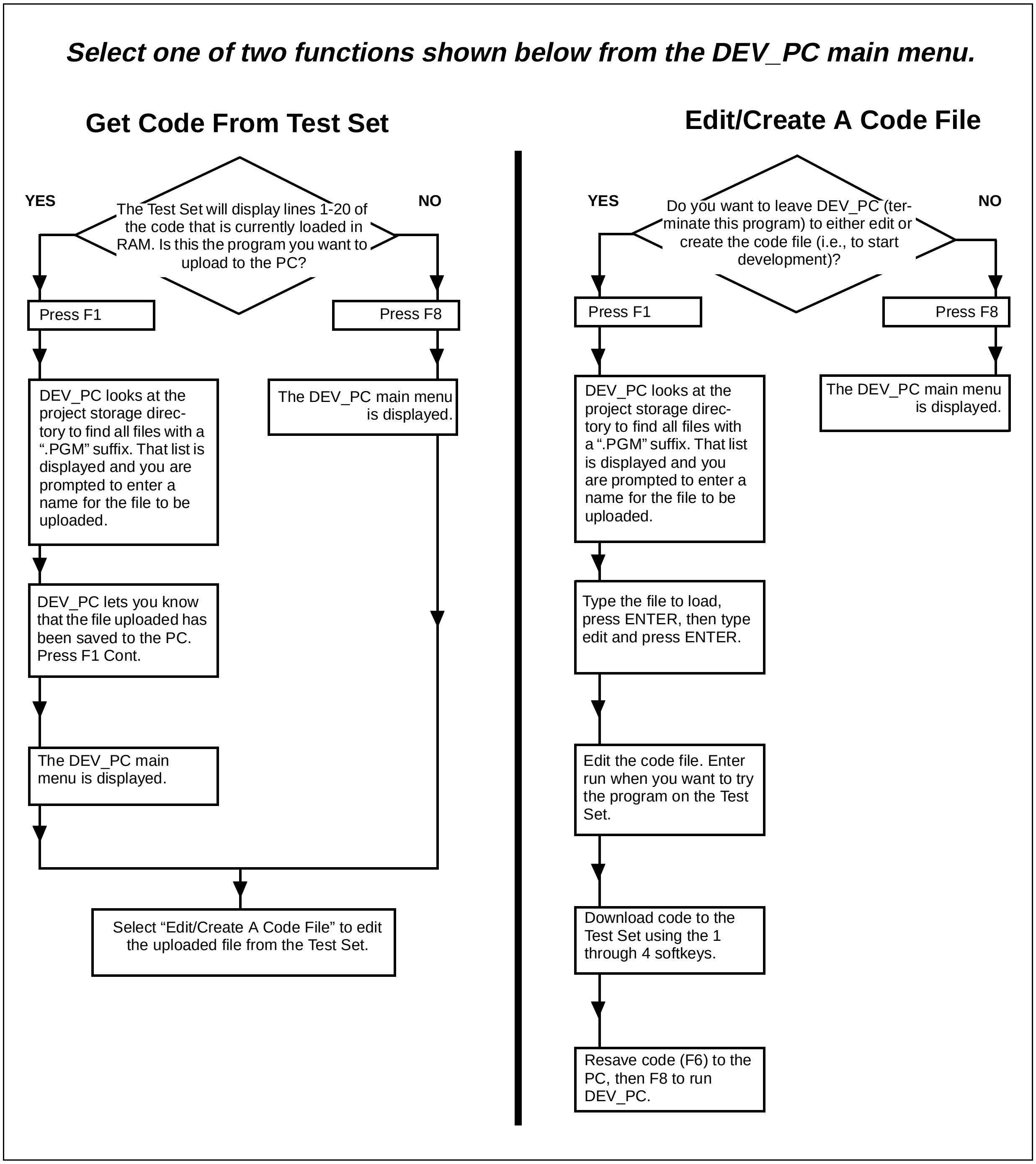

Select one of two functions shown below from the DEV_PC main menu.
Get Code From Test Set
YES
The Test Set will display lines 1-20 of the code that is currently loaded in RAM. Is this the program you want to upload to the PC?
NO
Press F1
Press F8
DEV_PC looks at the project storage directory to find all files with a ".PGM" suffix. That list is displayed and you are prompted to enter a name for the file to be uploaded.
The DEV_PC main menu is displayed.
DEV_PC lets you know that the file uploaded has been saved to the PC. Press F1 Cont.
The DEV_PC main menu is displayed.
Select "Edit/Create A Code File" to edit the uploaded file from the Test Set.
Edit/Create A Code File
YES
Do you want to leave DEV_PC (terminate this program) to either edit or create the code file (i.e., to start development)?
NO
Press F1
Press F8
DEV_PC looks at the project storage directory to find all files with a ".PGM" suffix. That list is displayed and you are prompted to enter a name for the file to be uploaded.
The DEV_PC main menu is displayed.
Type the file to load, press ENTER, then type edit and press ENTER.
Edit the code file. Enter run when you want to try the program on the Test Set.
Download code to the Test Set using the 1 through 4 softkeys.
Resave code (F6) to the PC, then F8 to run DEV_PC.

**Figure 8** **Editing or Creating Program Code at a Glance**

## Using the Tests Subsystem.

DEV_PC allows you to take advantage of the Tests Subsystem, which is the Test Set's automated user-interface. If you press the Test Set's TESTS key, you access the Tests Subsystem's main menu; at the **`TESTS (Main Menu)`** the sub-menus are selected in order to change test parameters, specifications, radio frequencies, and other test conditions that are normally hard-coded into a program.

The Tests Subsystem architecture works with three types of files: code, library, and procedure files. Hewlett-Packard has developed software specifically designed to run on the Test Set. The HP 11807 software provides the end-user with a library of industry-standard tests based on the published or manufacturers specifications and parameters. The end-user has the flexibility to change test frequencies/channels, specification limits, parameter values, and to control the testing environment. The relationship of these files is shown in figure 9 .

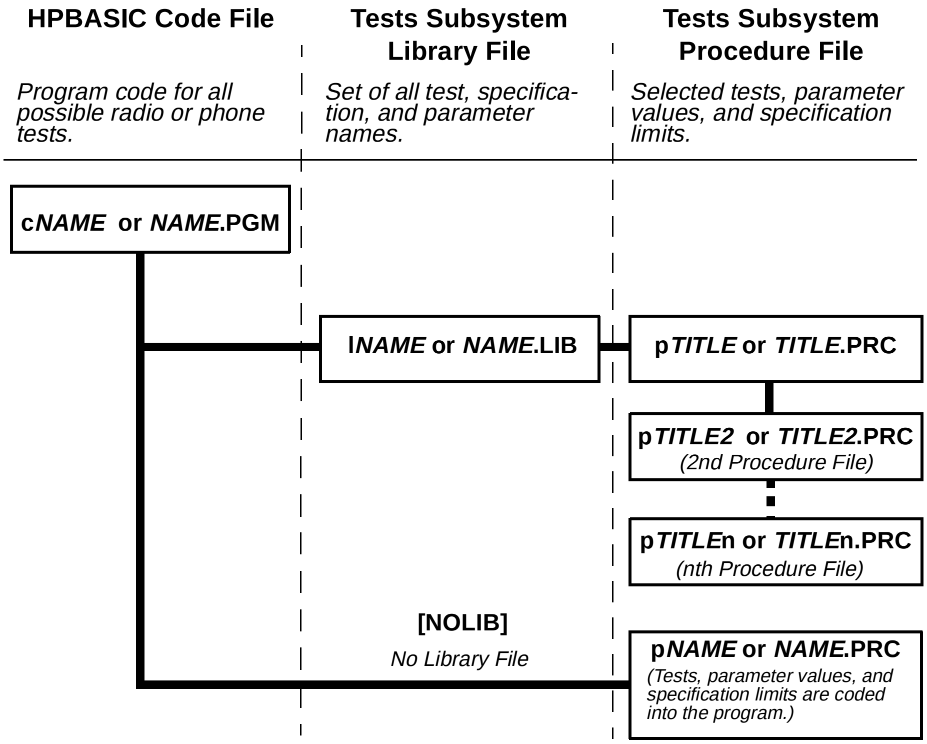

HPBASIC Code File
Program code for all possible radio or phone tests.
Tests Subsystem Library File
Set of all test, specification, and parameter names.
Tests Subsystem Procedure File
Selected tests, parameter values, and specification limits.
cNAME or NAME.PGM
lNAME or NAME.LIB
pTITLE or TITLE.PRC
pTITLE2 or TITLE2.PRC
(2nd Procedure File)
pTITLEn or TITLEn.PRC
(nth Procedure File)
[NOLIB]
No Library File
pNAME or NAME.PRC
(Tests, parameter values, and specification limits are coded into the program.)

**Figure 9** **Tests Subsystem File Relationship**

### Tests Subsystem File Descriptions

Let’s take a closer look at the three types of files that are used in the Tests Subsystem; each file is used to store different types of information.

### Code Files

The first aspect of an IBASIC program is the code itself. An IBASIC code file can reside on a memory card, on an external disk drive connected to the HP-IB port of the Test Set, or on the internal RAM disk of the Test Set. The name of this file is preceded by a lower case **`c`** in the HP 8920 and HP 8921 family of Test Sets, and a **`.PGM`** file extension in the HP 8920B. These file naming conventions tell the Tests Subsystem that this particular file contains program code.

### Library Files

A library file includes a list of the available test subroutines in the code, the set of all parameter names that are seen by using the Tests Subsystem’s sub-menu screens, and the set of all specification names that might be used by the subroutines in the code to decide if a test point passes or fails.

Only one library file is defined for each code file. The filename is preceded by a lower case **`l`** in the HP 8920 and HP 8921 family of Test Sets, and a **`.LIB`** file extension in the HP 8920B. These file extensions tell the Tests Subsystem that this particular file is a library file. Also, both the library and code file must have the same base name to indicate the relationship between them. For example, the code file **`cNAFM`** (or **`NAFM.PGM`**) must have the same library file relationship **`lNAFM`** (or **`NAFM.LIB`**) in order to work together within the Tests Subsystem.

A library file is required when you use the Tests Subsystem sub-menus; the library name, minus the lower-case **`l`** prefix or **`.LIB`** file extension, appears in the **`TESTS (Main Menu)`** next to the **`Library`** indicator. If the program is simple enough that there is no need for user-input, or if all the user-input is simple enough to be accomplished with INPUT statements, then a [NO LIB] option is available from DEV_PC (as shown in figure 9, "Tests Subsystem File Relationship," on page 32).

### Procedure Files

A procedure file allows the user to define parameter values, specification limits, and which of the test subroutines that will be used to test a specific radio or phone. There may be many procedure files defined that use the same IBASIC code and library files, each procedure using a different subset of the choices available in the library. The procedure files are preceded with a lower case **`p`** in the HP 8920 and HP 8921 Test Set family, and a **`.PRC`** file extension in the HP

8920B. Procedure files do *not* need the same base name as the case with a code and library file. The name of the corresponding library file (if any is used) is stored within the procedure file when it is created.

### When Should I Use the Tests Subsystem?

Programs that do not use the Tests Subsystem have the following limitations:

- Tests are always run in the same order, unless the program is written to allow change (as opposed to having the flexibility to choose the test order before testing, by using the Tests Subsystem submenu **`Seqn- Order of Tests`**).
- Different test procedures cannot be created from the same program without modifying the test program and re-storing it (as opposed to using the Tests Subsystem to create and store different procedures from the same program and library).
- If the program compares measured values to specifications, the same specifications must be used every time the test is run, or the program must be written to allow specification changes each time the test is run (as opposed to using the Tests Subsystem to change specification limits and store them with the associated procedure under a title you can remember for future use).
- If the operator wants to change instrument settings used in a test (frequencies, amplitudes, filters, etc.), the operator must either change the test program's variables directly, or the program must provide for operator interaction during the test (as opposed to changing these parameters before testing using the Tests Subsystem).
- Programs must be loaded and run directly from the **`TESTS(IBASIC Controller)`** screen, instead of using the **`Seqn- Order of Tests`** sub-menu.

Programs that do not use the Tests Subsystem have the following freedoms:

- Programs do not have to have a strictly defined structure.

The Tests Subsystem's capabilities were designed to allow the operator to "pick and choose" the tests and parameters needed from a larger set, thereby eliminating unnecessary tests and reducing test time. Writing programs to run in the TESTS environment requires the programmer to understand and adhere to the program structure and syntax required by the Tests Subsystem.

### Tests Subsystem Screens

The Tests Subsystem consists of the **`TESTS (Main Menu)`** screen, and several user-interface (submenu) screens. Continue reading to get a description of each of these screens.

### The TESTS (Main Menu) Screen

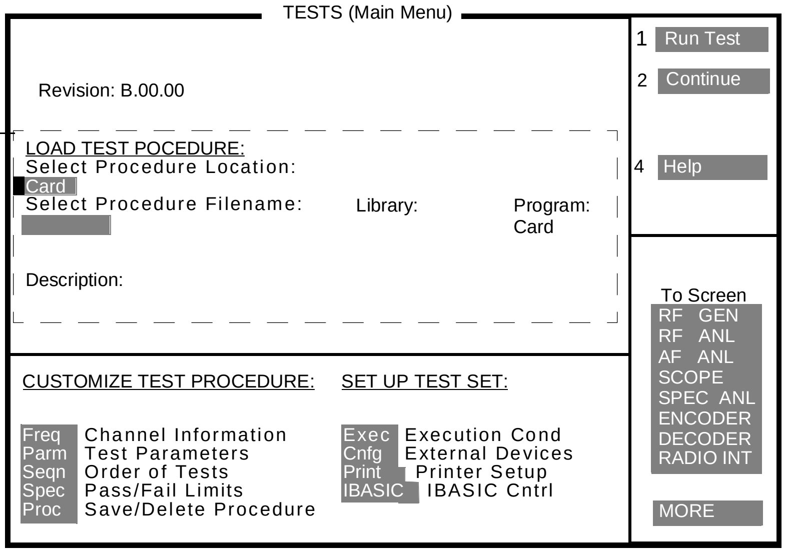
The **TESTS (Main Menu)** screen is accessed by pressing the Test Set's front panel TESTS key. Test procedures are selected and run from this screen. Additionally, access to all other Tests Subsystem screens is accomplished from this screen. Figure 10, "The TESTS Main Menu Screens," on page 36 illustrates the Test Set TESTS screens prior to firmware revision A.14.00 and after.

- The **Select Procedure Location:** or **Location** field is used to select the mass storage location for the procedure to be loaded.
- The **Select Procedure Filename:** or **Procedure:** field is used to select the name of the procedure to be loaded.
- The **Description:** or **Comment** field gives the user a brief description of the procedure currently selected in the **Select Procedure Filename** or **Procedure** field.
- The **Library** field indicates if the procedure uses a library, if not, **NO LIB** is displayed.

To view all the procedures available on the mass-storage location currently selected in the **Select Procedure Location:** field, position the cursor on the **Select Procedure Filename:** field and push the rotary knob. A menu will appear in the lower right corner of the screen, displaying all the procedure files which are available. This is not a listing of the full contents of the selected mass storage location, it is only a list of the procedures files saved at the mass-storage location corresponding to **pTITLE** or **TITLE.PRC** file format.

## TESTS Main Menu for Firmware A.14.00 and Above

Selects and loads a test procedure and its associated library file from the memory card to the internal memory of the Test Set.



TESTS (Main Menu)

Revision: B.00.00

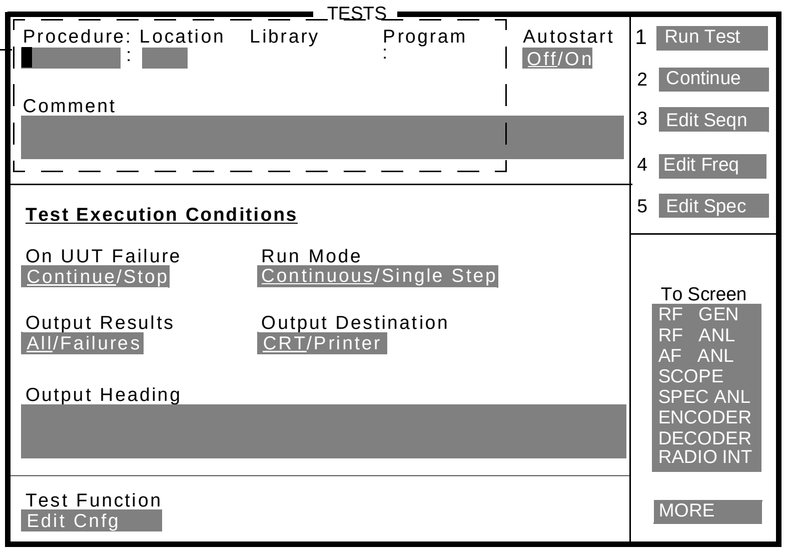
LOAD TEST POCEDURE:
Select Procedure Location:
Card
Select Procedure Filename:
Library:
Program:
Card
Description:

CUSTOMIZE TEST PROCEDURE:
Freq Channel Information
Parm Test Parameters
Seqn Order of Tests
Spec Pass/Fail Limits
Proc Save/Delete Procedure

SET UP TEST SET:
Exec Execution Cond
Cnfg External Devices
Print Printer Setup
IBASIC IBASIC Cntrl

## TESTS Main Menu for Firmware below A.14.00

Selects and loads a test procedure and its associated library file from the memory card to the internal memory of the Test Set.



TESTS

Procedure: Location Library Program
:
:
Autostart
Off/On
Comment

Test Execution Conditions

On UUT Failure
Continue/Stop

Run Mode
Continuous/Single Step

Output Results
All/Failures

Output Destination
CRT/Printer

Output Heading

Test Function
Edit Cnfg

**Figure 10** **The TESTS Main Menu Screens**

### Details about the Tests Subsystem Screens

The Tests Subsystem's main menu allows the user to easily modify the test subroutines, parameters, specifications and configuration settings to correspond to the requirements of a specific radio or phone. There are several user-interface screens (sub-menus) provided to allow the user to make modifications.

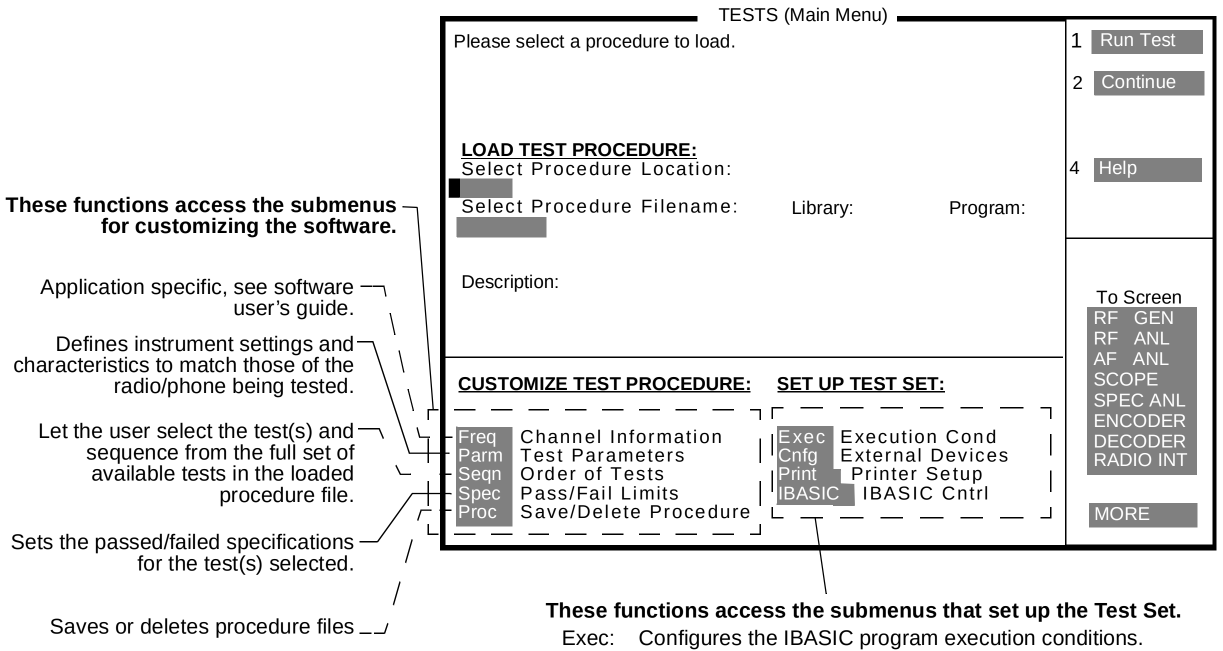
To access any of these screens, position the Test Set's cursor on the desired field and push the rotary knob. Refer to figure 11, "Tests Subsystem for Firmware A.14.00 and Above," on page 38 and figure 12, "Tests Subsystem for Firmware Below A.14.00," on page 39.

The following functions are available from the TESTS Main Menu (functions for firmware A.14.00 and newer are followed by the older version function names in parenthesis [below A.14.00]):

- The **Order of Tests** (or **Edit Sequence**) screen lets the user select the desired test(s) from the full set of available tests in the loaded procedure file.
- The **Channel Information** (or **Edit Frequencies**) screen defines the transmit and receive frequencies used for the selected tests.
- The **Pass/Fail Limits** (or **Edit Specifications**) screen defines the specifications used to generate pass/fail messages during testing.
- The **Test Parameters** (or **Edit Parameters**) screen is used to define instrument settings and characteristics to match those of the radio/phone being tested (audio load impedance, audio power, power supply voltage,..etc.).
- The **External Devices** (or **Edit Configuration**) screen identifies all connected HP-IB equipped instruments and their HP-IB addresses.
- The **Save/Delete Procedure** (or **Procedure Manager**) screen is used to save or delete Procedures.
- The **Printer Setup** (or **Test Execution Conditions**) screen is used to select the printer used for IBASIC PRINT commands and to configure the format of the printer page. Older versions of firmware do not have all of the printer setup settings.
- The **Execution Conditions** (or **Test Execution Conditions**) screen is used to configure the IBASIC program execution conditions.
- The **IBASIC Cntr**l (or **IBASIC**) screen is the IBASIC Controllers display screen.

Refer to the **TESTS** screen description in the Test Set's *User's Guide*, for information concerning how the different Tests Subsystem screens are used.

## TESTS Main Menu for Firmware A.14.00 and Above



TESTS (Main Menu)

Please select a procedure to load.

LOAD TEST PROCEDURE:
Select Procedure Location:
Select Procedure Filename: Library: Program:

Description:

CUSTOMIZE TEST PROCEDURE:

SET UP TEST SET:

**These functions access the submenus for customizing the software.**

Application specific, see software user’s guide.

Defines instrument settings and characteristics to match those of the radio/phone being tested.

Let the user select the test(s) and sequence from the full set of available tests in the loaded procedure file.

Sets the passed/failed specifications for the test(s) selected.

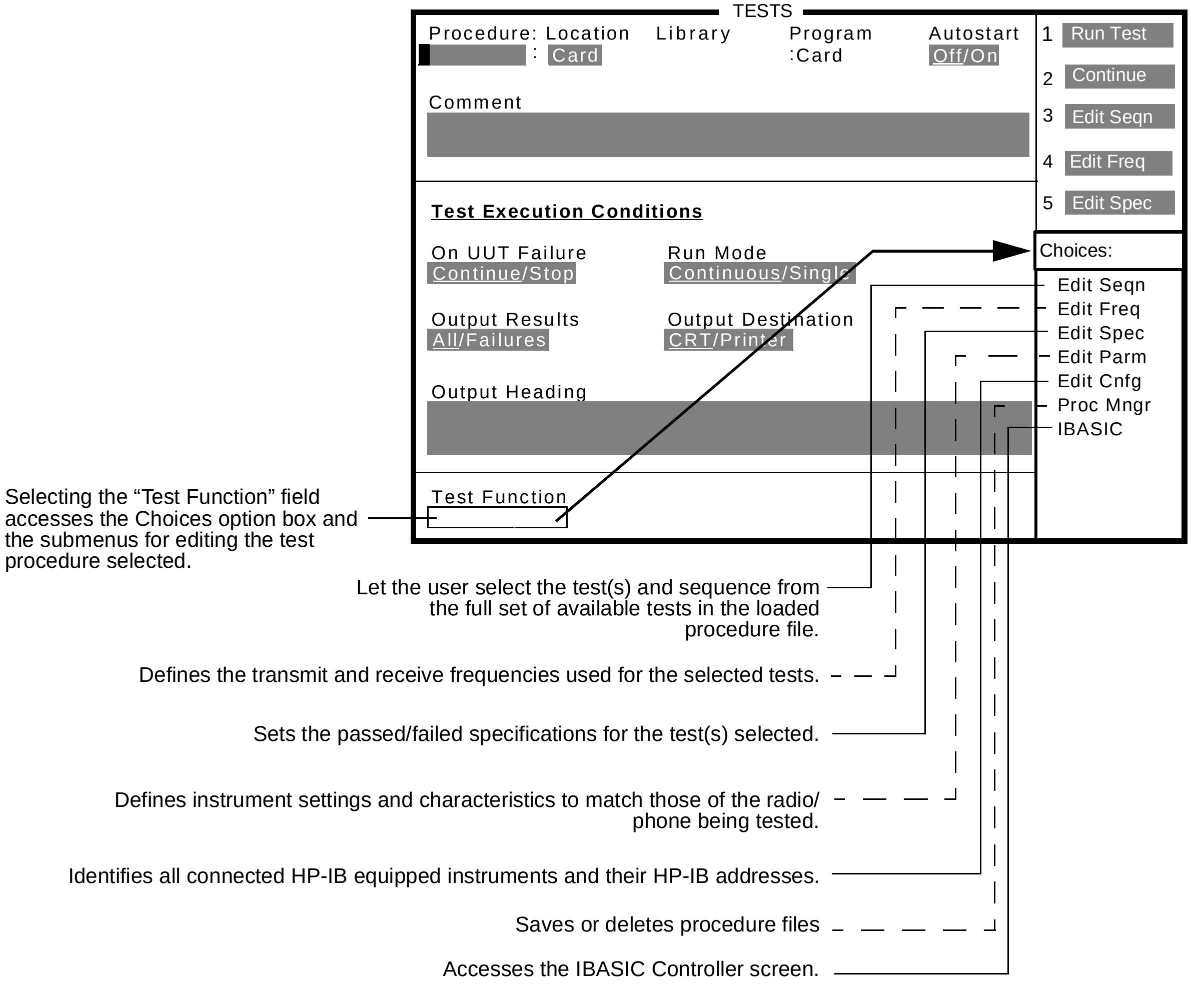
Saves or deletes procedure files

**These functions access the submenus that set up the Test Set.**

Exec: Configures the IBASIC program execution conditions.
Cnfg: Identifies all connected HP-IB equipped instruments and their HP-IB addresses.
Print: Selects the printer and configures the format of the printer page.
IBASIC: Accesses the IBASIC Controller screen.

**Figure 11** **Tests Subsystem for Firmware A.14.00 and Above**

## TESTS Main Menu for Firmware Below A.14.00



TESTS

Procedure: Location Library Program Autostart

: Card :Card Off/On

Comment

Test Execution Conditions

On UUT Failure Run Mode

Continue/Stop Continuous/Single

Output Results Output Destination

All/Failures CRT/Printer

Output Heading

Test Function

Selecting the “Test Function” field accesses the Choices option box and the submenus for editing the test procedure selected.

Let the user select the test(s) and sequence from the full set of available tests in the loaded procedure file.

Defines the transmit and receive frequencies used for the selected tests.

Sets the passed/failed specifications for the test(s) selected.

Defines instrument settings and characteristics to match those of the radio/ phone being tested.

Identifies all connected HP-IB equipped instruments and their HP-IB addresses.

Saves or deletes procedure files

Accesses the IBASIC Controller screen.

**Figure 12** **Tests Subsystem for Firmware Below A.14.00**

Figure 13 on page 41 shows how to interact with the DEV_PC main menu to create new, or modify existing library and procedure files. Have the DEV_PC main menu displayed on the PC and follow along.

A procedure file must have an associated library file or no-library selected in order for the file to appear in the list of choices when the user chooses the **`Select Procedure Filename`** field from the **`TESTS (Main Menu)`**.

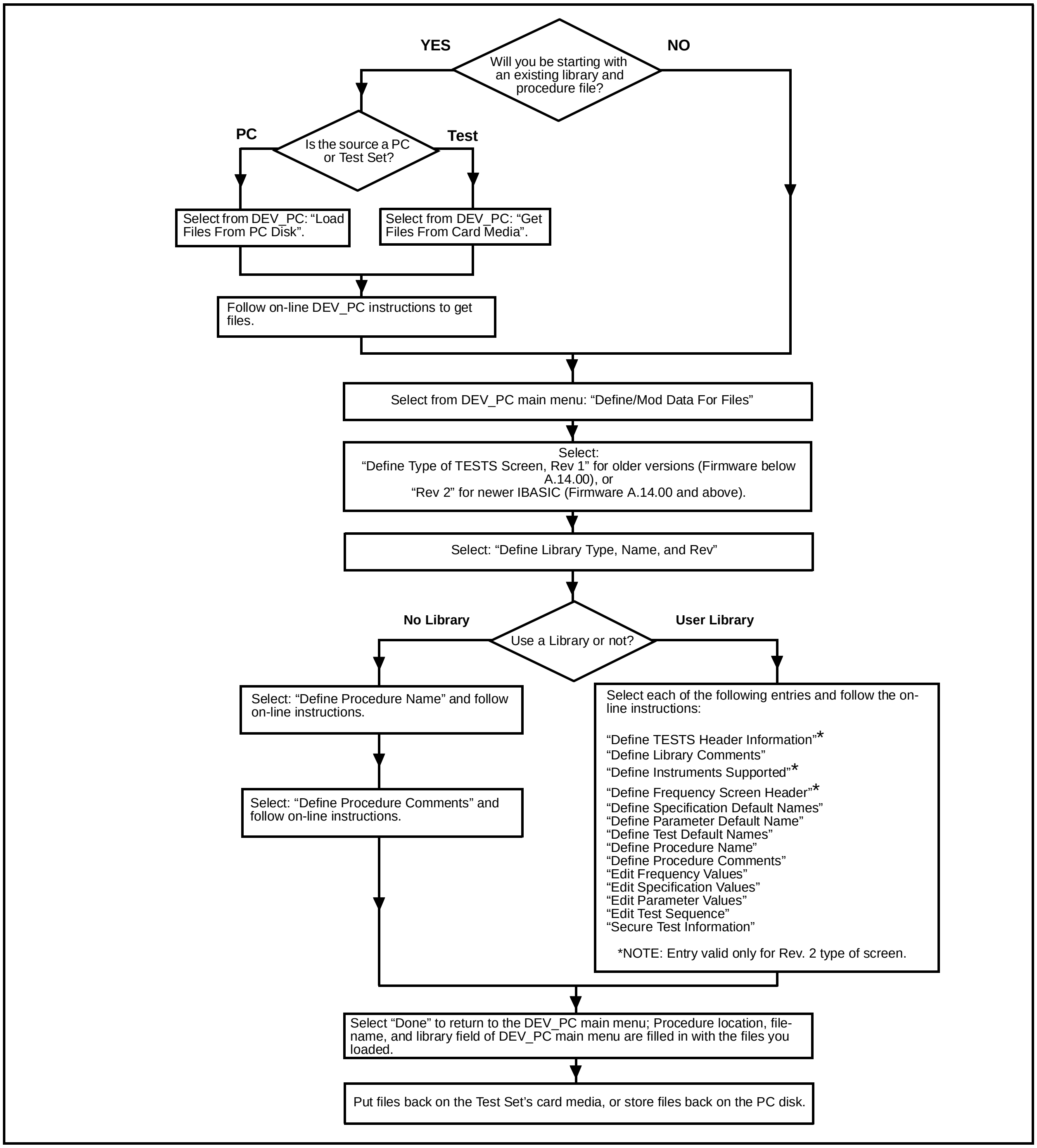

YES
Will you be starting with an existing library and procedure file?
NO
PC
Is the source a PC or Test Set?
Test
Select from DEV_PC: “Load Files From PC Disk”.
Select from DEV_PC: “Get Files From Card Media”.
Follow on-line DEV_PC instructions to get files.
Select from DEV_PC main menu: “Define/Mod Data For Files”
Select:
“Define Type of TESTS Screen, Rev 1” for older versions (Firmware below A.14.00), or
“Rev 2” for newer IBASIC (Firmware A.14.00 and above).
Select: “Define Library Type, Name, and Rev”
No Library
Use a Library or not?
User Library
Select: “Define Procedure Name” and follow on-line instructions.
Select: “Define Procedure Comments” and follow on-line instructions.
Select each of the following entries and follow the on-line instructions:
“Define TESTS Header Information”*
“Define Library Comments”
“Define Instruments Supported”*
“Define Frequency Screen Header”*
“Define Specification Default Names”
“Define Parameter Default Name”
“Define Test Default Names”
“Define Procedure Name”
“Define Procedure Comments”
“Edit Frequency Values”
“Edit Specification Values”
“Edit Parameter Values”
“Edit Test Sequence”
“Secure Test Information”
*NOTE: Entry valid only for Rev. 2 type of screen.
Select “Done” to return to the DEV_PC main menu; Procedure location, file-name, and library field of DEV_PC main menu are filled in with the files you loaded.
Put files back on the Test Set’s card media, or store files back on the PC disk.

**Figure 13** **Create or Modify Library and Procedure Files at a Glance**

## Converting Files from HP 8920 and HP 8921 for Use on the HP 8920B

DEV_PC allows custom developed software written for use on the HP 8920 and HP 8921 test sets to be converted for use on an HP 8920B. With DEV_PC you can:

1. Change the software's file format; for example change it from LIF to DOS or visa versa.
2. Change the memory card type; for example change it from EPSON to PCMCIA or visa versa.

Figure 14 on page 43 describes how to use DEV_PC to convert LIF formatted files from an HP 8920 or HP 8921, to a DOS formatted PCMCIA memory card on the HP 8920B. It is done with the assumption that code, library, and procedure files already exist and are either resident in the PC or on a memory card in the Test Set.

***NOTE:*** After converting LIF formatted files from an HP 8920 or HP 8921 to the DOS format, you will still need to debug the code for issues related to timing (since the HP 8920B Test Sets runs faster) and for errors that result from model-number options on the HP 8920 or HP 8921 Test Sets that are not recognized by the HP 8920B.

### Test Set File Types

The filing system for the Test Sets support the following file types, which are seen when you catalog a memory card:

- ASCII - files with data in LIF ASCII format
- BDAT - files containing binary data
- DIR - DOS subdirectory
- DOS - saved or stored code files
- HP-UX - stored code file (HP 8920 or HP 8921 Test Sets only)
- IBPRG - stored code file (HP 8920B Test Set only)

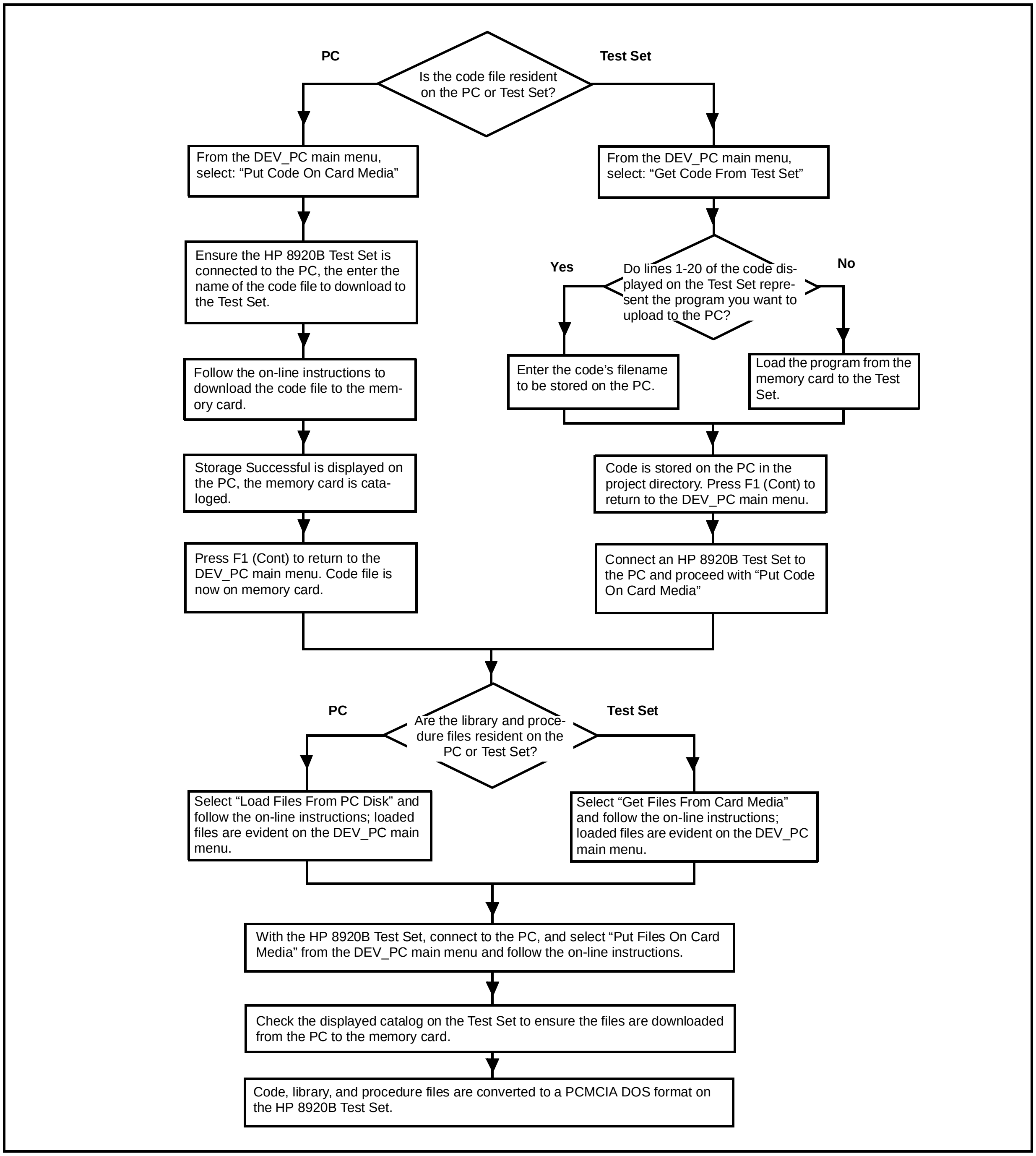

PC
Is the code file resident on the PC or Test Set?
Test Set
From the DEV_PC main menu, select: “Put Code On Card Media”
Ensure the HP 8920B Test Set is connected to the PC, the enter the name of the code file to download to the Test Set.
Follow the on-line instructions to download the code file to the memory card.
Storage Successful is displayed on the PC, the memory card is cataloged.
Press F1 (Cont) to return to the DEV_PC main menu. Code file is now on memory card.
From the DEV_PC main menu, select: “Get Code From Test Set”
Yes
Do lines 1-20 of the code displayed on the Test Set represent the program you want to upload to the PC?
No
Enter the code’s filename to be stored on the PC.
Load the program from the memory card to the Test Set.
Code is stored on the PC in the project directory. Press F1 (Cont) to return to the DEV_PC main menu.
Connect an HP 8920B Test Set to the PC and proceed with “Put Code On Card Media”
PC
Are the library and procedure files resident on the PC or Test Set?
Test Set
Select “Load Files From PC Disk” and follow the on-line instructions; loaded files are evident on the DEV_PC main menu.
Select “Get Files From Card Media” and follow the on-line instructions; loaded files are evident on the DEV_PC main menu.
With the HP 8920B Test Set, connect to the PC, and select “Put Files On Card Media” from the DEV_PC main menu and follow the on-line instructions.
Check the displayed catalog on the Test Set to ensure the files are downloaded from the PC to the memory card.
Code, library, and procedure files are converted to a PCMCIA DOS format on the HP 8920B Test Set.

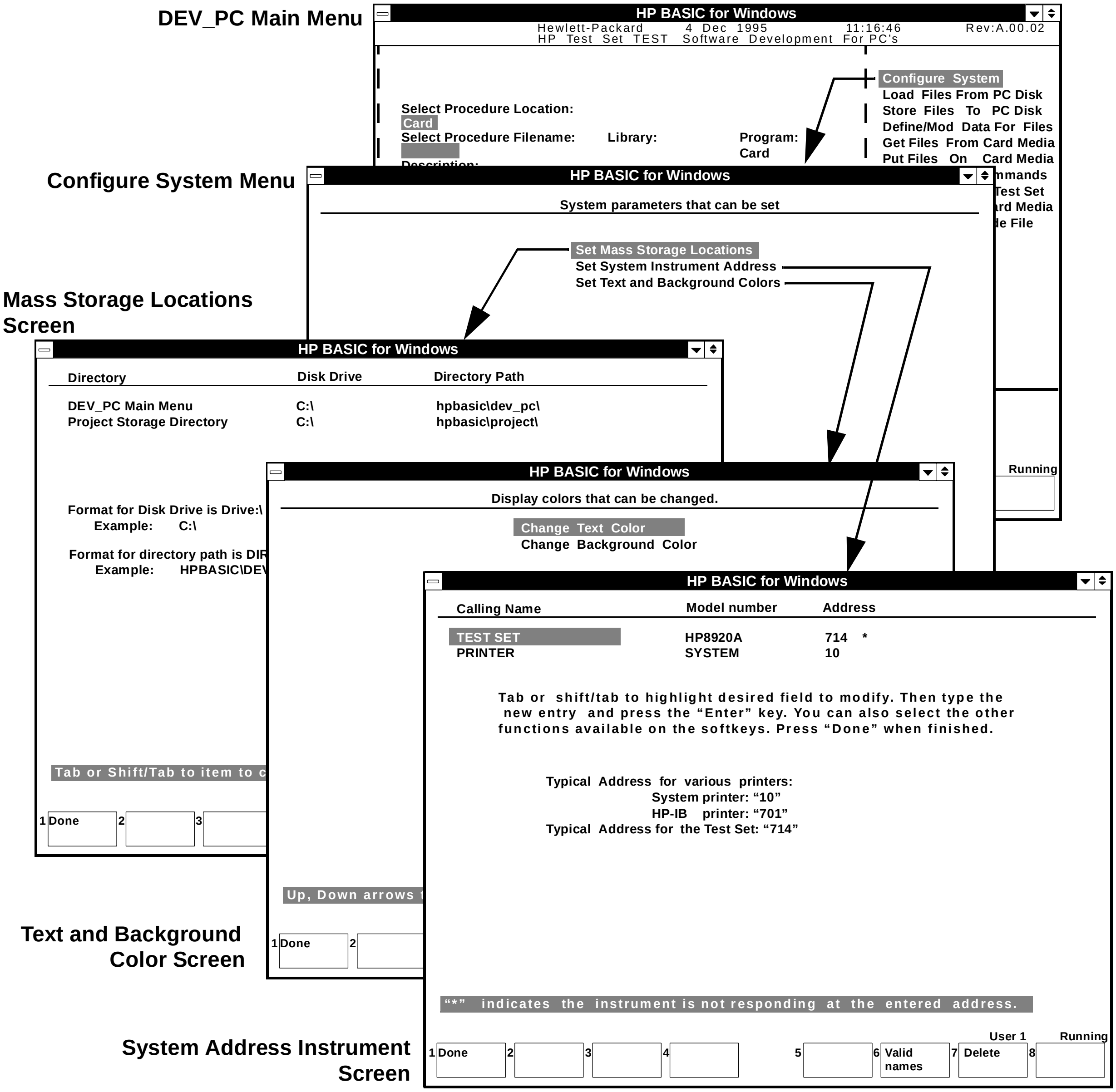
**Figure 14** Converting HP 8920 and HP 8921 Code, Library and Procedure Files at a Glance

## Take a Closer Look at the DEV_PC "Configure System"

This section covers three important system parameters of DEV_PC:

- Setting mass-storage locations
- Setting instrument addresses for equipment in the system
- Changing text and background colors

Figure 15 on page 45 shows you how to interact with the DEV_PC main menu to change any of the system parameters mentioned above.



Hewlett-Packard 4 Dec 1995 11:16:46 Rev:A.00.02
HP Test Set TEST Software Development For PC's

Select Procedure Location:
Card
Select Procedure Filename: Library: Program:
Card
Description:

System parameters that can be set

| Directory | Disk Drive | Directory Path |
|---|---|---|
| DEV_PC Main Menu | C:\ | hpbasic\dev_pc\ |
| Project Storage Directory | C:\ | hpbasic\project\ |

Format for Disk Drive is Drive:\
Example: C:\

Format for directory path is DIR
Example: HPBASIC\DEV

Tab or Shift/Tab to item to c

Display colors that can be changed.

Up, Down arrows t

| Calling Name | Model number | Address |
|---|---|---|
| TEST SET | HP8920A | 714 * |
| PRINTER | SYSTEM | 10 |

Tab or shift/tab to highlight desired field to modify. Then type the new entry and press the “Enter” key. You can also select the other functions available on the softkeys. Press “Done” when finished.

Typical Address for various printers:
System printer: “10”
HP-IB printer: “701”
Typical Address for the Test Set: “714”

“*” indicates the instrument is not responding at the entered address.

Figure 15 DEV_PC Configuration System Functions

## Mass Storage Locations Screen

Use the Tab or Shift/Tab keys to set up the Disk Drive and Directory Path Locations.

Select the 1(Done) softkey and choose Yes or No to store the changes you made on the PC. The changes are saved in a file under the DEV_PC main directory that you set up when you selected "Set Mass Storage Locations". The file's names is "mass_pc".

Disk drive selections are limited to those on the PC, or those locations on a network. HP-IB addresses for disk drives are not valid.

## System Instrument Address Screen

Use the Tab or Shift/Tab to set up the model and address of the devices on the PC. Only two choices are valid: Test Set and Printer. Printer can be on the system (PC) or on HP_IB.

Select the 1 (Done) softkey and choose Yes or No to store the changes you made on the PC. The changes are saved in a file under the DEV_PC main directory that you set up when you selected "Set Mass Storage Locations". The file's names is "inst_pc".

An asterisk next to the Address indicates that the device is not turned on or connected. Refer to the HP BASIC Help icon under I/O Device Drivers if you need more information to configure a printer to your PC.

## Set Text and Color Screen

Use the Up/down arrow keys to select "Change Text Color" or "Change BackGround Color" option. Press the ENTER key to make changes.

Select the 1 (Done) softkey and choose Yes or No to store the changes you made on the PC. The changes are saved in a file under the DEV_PC main directory that you set up when you selected "Set Mass Storage Locations". The file's names is "color_pc".

Color settings are not turned on when HP BASIC for Windows is first loaded, however, upon selecting Exit from the DEV_PC main menu, colors are used.

5

# Basics of Code, Library, and Procedure Files

This chapter provides basic instructions to use DEV_PC for developing code, and for modifying and creating library and procedure files.

# Basics of Developing Code Using the Tests Subsystem

The following exercise is a detailed example designed to instruct you through the steps required to create a program on DEV_PC and run it from a memory card using the Tests Subsystem. The exercise uses a simple program; you may choose to substitute it with another if you desire. Being that the program is a simple one, it does not require the use of test specifications or parameters, so the exercise instructs you to create a [NO LIB] procedure. When you are done, you can view the procedure on the memory card from the **`TESTS (Main Menu)`**.

***NOTE:*** The *HP Instrument BASIC User's Handbook* has information essential to the success of your programming. Use this handbook to find information about programming techniques, interfacing techniques, and BASIC language reference.

Also, refer to the *HP 8920 and HP 8921 Programmer's Guide* for specific information about programming the Test Set. Information specific to HP-IB, advanced operation techniques, memory cards, and the IBASIC controller are not fully covered in this manual.

## Example to Develop Code

The following example to create, run, and download a code file to a memory card takes about 1 hour to complete.

### Create a Simple Program

1. Insert a memory card in the Test Set and make sure it is not set to the Write Protect position.
2. Run HP IBASIC for Windows.
3. Get and run DEV_PC. (Refer to *Getting Started* in this manual if you have any problems running DEV_PC.)
4. Select **`Edit/Create A Code File`** from the DEV_PC main menu, and press F1 to terminate the DEV_PC program to start code development.

***NOTE:*** You could choose to write the simple program directly in HP BASIC for Windows window, however, you would not have use of the 1 through 4 download softkeys.

5. Clear the command line of the **`GET`** statement (unless you saved the simple program, shown below, from the installation procedure), type **`scratch`** and press the Enter key, then type **`edit`** and press the Enter key. Type in the following:

```
10 !FREQ ! Give the program a name, it must appear on the 1st line of code.
20 Uut=714 !Test Set’s address to be used by PC for control
30 OUTPUT Uut;”*rst”
40 WAIT 2
50 OUTPUT Uut;”rfg:freq 123 mhz” !sets RF generator to 123 MHz
60 OUTPUT Uut;”rfg:ampl -20 dbm” !sets RF amplitude to -20 dBm
70 OUTPUT Uut;”disp dupl” !sets up the Duplex Test screen
80 OUTPUT Uut;”meas:rfr:freq:abs?” !query RF generator
90 ENTER Uut:A !put value in “A”
100 PRINT “Frequency is:”;A !Print value of “A” to PC or Test Set
110 END
```

### Run Simple Program and Save it to PC

**6.** Type **`run`** and press the Enter key. The program should display the RF generator's frequency on the PC; if not, then re-check the program to make sure it is typed in correctly.

**7.** Change line 20 to HP-IB address 800. (The address at line 20 may be different than the suggested factory-default of 714, but, an HP-IB address of 800 allows the program to run on the Test Set's internal HP-IB bus.)

**8.** Select softkeys 1 through 4 in order to download the program to the Test Set. (These softkeys perform the necessary functions to download code from the PC to the Test Set without having to make you type in each separate command in the proper sequence.)

**9.** Press the LOCAL key on the Test Set to take it out of remote control. Make sure that the **`TESTS IBASIC Controller`** screen is displayed then press the k1 **`Run`** user key; the RF generator's frequency should be displayed on the Test Set.

**10.** Select the 6 softkey **`RE-SAVE to Disk`** to save the simple program to the PC.

**11.** Select the 8 softkey to run DEV_PC.

### Create a [NO LIB] Procedure

**12.** At the DEV_PC main menu, select **`Define/Mod Data For Files`** option.

**13.** Select **`Define Type of TESTS Screen`**, have **`Rev 1`** or **`Rev 2`** selected according to the on-line instructions, then select 1 **`Done`** softkey.

**14.** Select **`Define Library Type, Name and Rev`**, then select 5 **`Select`** to choose **`NO LIBRARY`**, finally select 1 **`Done.`**

**15.** Select **`Define Procedure Name`**, and enter **`FREQ`** and press the Enter key when you're done and F1 **`Yes`** if the name is correct.

***NOTE:*** The filename you enter can be up to 8 characters in length; in the previous step you were asked to enter FREQ as the filename since it was the name put in the first line of code on line 10. For a [NO LIB] procedure, the first line of code must contain the procedure's filename commented out.

**16.** Select **`Define Procedure Comments`** and type in up to 2 lines of comments which will be displayed on the **`TESTS (Main Menu)`** when the procedure is loaded into the Test Set, then select 1 **`Done`** softkey.

**17.** Select 1 **`Done`** to return to the DEV_PC main menu. The procedure information you entered in the previous four steps will be displayed in the upper left-half of the DEV_PC main menu.

### Download Code and Procedure Files to the Test Set's Memory Card

**18.** At the DEV_PC main menu, select **`Put Code on Card Media`**., then enter the name of the code file to download, and press the Enter key and follow on-line instructions. (The code file you downloaded should appear on the Test Set when it is cataloged after the download procedure.)

**19.** At the DEV_PC main menu, select **`Put Files on Card Media`**, and follow the on-line instructions.

***NOTE:*** At this point in the exercise, select F1 **`Yes`** to continue. A program titled **`XFER.PGM`** is loaded into the Test Set to handle the transfer of the library and/or procedure files to the memory card. Any code currently in the Test Set is overwritten.

**20.** Check the Test Set display to make sure the downloaded library and/or procedure files were transferred from the PC, press F1 **`Cont`** if they are on the cataloged memory card.

**21.** Press the LOCAL key on the Test Set , press the TESTS key, and with **`CARD`** selected as the procedure location, select the Test Procedure you saved. On the Test Set, you should see: the procedure name you made on the PC, any comments you entered, and [NO LIB] below the library field.

**22.** Press the k1 **`Run`** key to test the results of the program you downloaded.

***NOTE:*** Continue to the exercise section "Basics of Modifying and Creating Library and Procedure Files" on page 52 if you want to learn more about procedures that use a library.

# Basics of Modifying and Creating Library and Procedure Files

There are two exercises to this section:

- ❐ *Modify a Library and Procedure File* (Do only if you have access to an HP 11807 software program, or have access to a program written to use the Tests Subsystem).
- ❐ *Create a New Library and Procedure File* (Do by creating a simple program and then creating a new library and procedure file.)

In the process of completing this section, you will have learned how to save library and procedure files to a memory card, and how to verify that the files work on the Test Set.

## Modify a Library and Procedure File

Do not start this procedure unless you have a copy of HP 11807 software or some other software that was written following the Tests Subsystem architecture . If you don’t have access to Tests Subsystem based software, turn to the next procedure titled "Create a New Code, Library and Procedure File" on page 56.

### Up load the Library and Procedure Files into the PC

1. At the DEV_PC main menu, select one of the following choices depending if the library and procedure files are on the PC hard-disk drive, or on the Test Set’s memory card:

   **`Load Files From PC Disk`**

   **`Get Files From Card Media`**

   In either case, follow the on-line instructions to upload the files. The DEV_PC main-menu display verifies that the library and procedure files have been loaded.

### Access the Area where Library and Procedure Files are Modified

2. Select **`Define/Mod Data For Files`** at the DEV_PC main menu.
3. When library and procedure files are uploaded to the PC, certain pieces of information about the files should have been already defined; so, depending on your needs, you may or may not need to make changes to the following areas:

```
Define Type of TESTS Screen
Define Library Type, Name and Rev
Define TESTS Header Information
Define Library Comments
```

**Modifying the Library**

4. Select **`Define Instruments Supported`**. (The changes you make to this area are reflected on the Test Set when you select **`External Devices`** from the **`TESTS (Main Menu)`**. When you move the cursor and single-click on the **`Calling Name field`** , you'll get a list of choices defined by this area.)

5. Select **`Define Frequency Header Screen`**. (The changes you make to this area are reflected on the Test Set when you select **`Frequency Information`** from the **`TESTS (Main Menu)`**.

6. Select **`Define Specification Default Names`**. (The changes you make to this area are reflected on the Test Set when you select **`Pass/Fail Limits`** from the **`TESTS (Main Menu)`**. Modify the specification names and the units that the pass/fail limits will be specified in. Up to 50 total specifications and units may be entered.

7. Select **`Define Parameter Default Names.`** (The changes you make to this area are reflected on the Test Set when you select **`Test Parameters`** from the **`TESTS (Main Menu)`**. Modify the parameter names and their units. Up to a total of 50 parameters and units may be entered.

8. Select **`Define Test Default Names.`** (The changes you make to this area are reflected on the Test Set when you select **`Order of Tests`** from the **`TESTS (Main Menu)`**. Modify the test names; the gosub name for each test is displayed in a column to the left of each test name. Up to a total of 50 tests may be entered.

### Modifying the Procedures

**9.** Select **`Define Procedure Name.`** Enter a name to be assigned to the procedure currently being worked on. The name you assign (up to 8 characters in length) appears in the **`TESTS (Main Menu)`** and the DEV_PC main menu's **`Select Procedure Filename`** field.

**10.** Select **`Define Procedure Comments`**. Enter up to 2 lines of comments that appear in the **`TESTS (Main Menu)`** and the DEV_PC main menu's **`Description`** area.

**11.** Select **`Edit Frequency Values.`** Enter up to 50 frequency pairs and their associated **`Test?`** and **`Prime?`** settings. The frequencies and channel information you enter in this area appears in the **`Channel Information`** menu in the **`TESTS (Main Menu)`**.

The query **`Test?`** implies that the channel will or will not be tested; the query **`Prime?`** is associated with the query **`All Chans?`** in the **`Order of Tests`** menu. (For example, if you decide that "yes" the channel will be tested and that it is "yes" a prime channel, then by selecting a test to run on **`All Chans`** "yes" , that channel will be used by that test every time.)

**12.** Select **`Edit Specification Values.`** Enter the default upper and lower specification limits to be used by the procedure. The specification limits you enter in this area appears in the **`Pass/Fail Limits`** menu in the **`TESTS (Main Menu)`**.

**13.** Select **`Edit Parameter Values`**. Enter the default parameter limits to be used by the procedure. The parameter limits you enter in this area appears in the **`Test Parameters`** menu in the **`TESTS (Main Menu)`**.

**14.** Select **`Edit Test Sequence`**. Select the default tests you want to be used by the procedure. The choices you make appear in the **`Order of Tests`** menu in the **`TESTS (Main Menu)`**. A test can be selected as a prime channel (to be used by the channels also selected as "prime", or the test can be selected to run on all channels.

**15.** Select **`Secure Test Information`**. If you want to "secure" the test sequence, frequencies, specifications, parameters, or system configuration, you must enter a password. Setting a password prevents un-authorized modifications of the procedure file.

### Download Library and Procedure Files back to PC and Test Set

**16.** Select **`Store Files To PC Disk`**. and follow the on-line instructions to store files back onto the PC hard-disk (or network drive).

**17.** Select **`Put Files On Card Media`** and follow the on-line instructions to download the library and procedure files on the Test Set's memory card.

***NOTE:*** With the modified library and procedure files back on the memory card, make sure that there is a fully charged battery installed in the card. Note on the card when it was installed, since batteries have a shelf life of about 1 year.

## Create a New Code, Library and Procedure File

The following example instructs you to first create a library with three simple parameters and a procedure to enter values for each of the three parameters. Then you write a simple program to perform arithmetic calculations using the three parameter values.

The steps involved in this example instruct you to save the code, library, and procedure files to a memory card in the Test Set. You will load and run the program from the memory card, change the parameter values on the Test Set and then re-run the program to ensure that the changes are acknowledged by the program.

*NOTE:* The principles learned in the following example are useful in understanding how to develop more complicated programs that use the Tests Subsystem.

### Create a Library File for the Calculator Program

1. At the DEV_PC main menu, select **`Define/Mod Data For Files`**.
2. Select **`Define Type of TESTS Screen`** and select **`Rev 1`** or **`Rev 2`** depending upon the firmware revision in your Test Set.
3. Select **`Define Library Type, Name and Rev`** and then select **`USER LIBRARY`**:

   **a** At this point, enter a name for the library. Use **`CALC`** as the library name. The name refers to "calculator", which is descriptive of what is done in this exercise

   **b** Enter the library revision, today's date. For example **`DEC 15 95`**.

   **c** Notice the library name and revision date appears at the top of the PC display when you return to the **`Define/Mod Data For Files`** menu.
4. (Skip this step if your Test Set does not have firmware A.14.00 or newer!) Select **`Define TESTS Header Information`** and type in "Simple Calculator", and keep **`Revision: A.00.00`** on the second line.

   With firmware revision A.14.00 and newer, the Test Set displays what you type, in this case "Simple Calculator", at the top of the DEV_PC main menu, as well as at the top of the **`TESTS (Main Menu)`** screen when the library file is downloaded into the Test Set.

5. In the **`Define/Mod Data For Files`** menu, skip the following User Library selections, they are not needed for the purpose of this exercise; however, you may want to use them for an application outside of this exercise:

```
Define Library Comments
Define Instruments Supported
Define Frequency Screen Header
Define Specification Default Names
Define Test Default Names
Edit Frequency Values
Edit Specification Values
Edit Test Sequence
Secure Test Information
```

***NOTE:*** Refer to the *Reference Information* section of this manual for further information about the preceding User Library selections.

6. Select **`Define Parameter Default Names`** and type in the following for parameter numbers 1 through 3:

```
1 Operand A
2 Operand B
3 Multiply=0 and Add=1
```

### Create a Procedure File for the Calculator Program

7. Select **`Define Procedure Name`** and enter any eight-character name to describe the procedure file being created; for example, **`ADD_MULT`**. (The procedure name appears on the Test Set in the **`TESTS (Main Menu)`** screen when the procedure is selected off the memory card.)

Notice the procedure name appears at the top of the PC display when you return to the **`Define/Mod Data For Files`** menu.

8. Select **`Define Library Comments`** and type in up to two lines of comments to describe what the program will accomplish. (The comments appear on the **`TESTS (Main Menu)`** in the Test Set when the library file is downloaded.) For example, you may type:

```
This is a simple calculator program.
Three parameters are available for you to use.
```

9. Select **`Edit Parameter Values`** and enter the following values:

```
Parameter value
5
10
0
```

### Download the Library and Procedure Files to the Test Set

**10.** At the DEV_PC main menu, select **`Put Files On Card Media`** and follow the on-line instructions. You will be prompted to “check the displayed catalog on the Test Set” for both the library and procedure files.

**11.** On the Test Set, press the LOCAL key, and the TESTS key.

**12.** Choose **`CARD`** as the procedure location, then select the procedure filename **`ADD_MULT`**. Notice that the procedure comments and other information you previously entered is displayed in the **`TESTS (Main Menu)`**.

***NOTE:*** An **`ERROR 56 File name is undefined`** appears on the Test Set if you attempted to run the procedure without the code file loaded into the Test Set at this time of the exercise.

### Create the Code File for the Calculator Program

**13.** At the DEV_PC main menu, select **`Edit/Create A Code File`** .

**14.** When prompted by the on-line instructions to “Type name of file to load”, type **`SCRATCH`** and press the Enter key, and then type **`EDIT`** and press the Enter key.

**15.** Type the following calculator program:

```
10 !CALC !Enter the library filename on this line
20 CLEAR SCREEN !Clears the display after each run
30 DIM A$[100] !Dimension array for parameter 1
40 DIM B$[100] !Dimension array for parameter 2
50 DIM C$[100] !Dimension array for parameter 3
60 Utt=714 !Enter the Test Set’s HP-IB address
70 OUTPUT Uut;”TESTS:PARM? 1” !Returns parameter 1 value
80 ENTER Uut;A$
90 X=VAL(TRIM$(A$[4]))
100 PRINT “Parameter 1 =”;X !Prints value of parameter 1
110 OUTPUT Uut;”TESTS:PARM? 2” !Returns parameter 2 value
120 ENTER Uut;B$
130 Y=VAL(TRIM$(B$[4]))
140 PRINT “Parameter 2 =”;Y !Prints value of parameter 2
150 OUTPUT Uut;”TESTS:PARM? 3” !Returns parameter 3 value
160 ENTER Uut;C$
170 Z=VAL(TRIM$(C$[4]))
180 IF Z=0 THEN
190 PRINT “Total =”;X*Y !Print parameter 1 times parameter 2
200 ELSE
210 PRINT “Total =”;X+Y !Print parameter 1 plus parameter 2
220 END IF
230 END
```

**16.** Run the calculator program from the PC. The PC display should show the following:

```
Parameter 1= 5
Parameter 2= 10
Total = 50
```

**17.** Edit line 60 of the program to change the HP-IB address from 714 to 800 (which is the internal HP-IB address of the Test Set).

**18.** Select the F6 softkey to re-save the code program to the PC giving it the name **`CALC`**.

### Download the Calculator program to the Test Set

**19.** Select softkeys F1 through F4 in order. (The DEV_PC softkeys 1 through 4 performs the necessary IBASIC commands to download the program to the Test Set.)

**20.** At the Test Set, select the LOCAL key and then select the K1 USER key to run the program. You should observe the same results as those when the program was run on the PC.

***NOTE:*** At this point in the exercise, the calculator program resides in the Test Set's RAM memory. Continue on with this exercise to download the program into the Test Set's memory card.

### Download the Calculator Program into the Memory Card

**21.** On the PC, select the F8 softkey to run DEV_PC.

**22.** Select **`Put Code On Card Media`** from the DEV_PC main menu.

**23.** Select the code file **`CALC`** and follow the on-line instructions to finish the download process to the memory card. The code file should be displayed on the Test Set and the PC should indicate that storage was successful.

### Run the Calculator Program from the Memory Card

**24.** Put the Test Set in the local mode and press the TESTS key.

**25.** Select the **`Test Parameters`** menu (**`Edit Parm`** for Test Set firmware older than A.14.00) and change parameters 1 through 3.

**26.** Press the K1 USER key to re-run the program. Notice the changes you made to parameters 1 through 3 are reflected in the new test results!

***NOTE:*** You may want to select the IBASIC function on the Test Set and catalog (**`CAT`**) the memory card to see the code, library, and procedure files for this exercise.

# 6

# Reference Information

This chapter provides reference information about the following: the files found on the DEV_PC floppy disk; the library and procedure file architecture; DEV_PC functional descriptions; a sample IBASIC program with code comments.

# Files on the DEV_PC Disk

The following five files must be stored under the same directory; for example you may have the following directory paths set up for DEV_PC and the project directory where you'll keep user-created code, library and procedure files:

```
c:\hpbasic
      |---- \dev_pc
      |         |---- dev_pc
      |         |----  inst_pc
      |         |---- mass_pc
      |         |---- color_pc
      |         |---- xfer.pgm
      |
      |----\project1
```

**dev_pc**

This file contains the IBASIC Software Development program DEV_PC. The revision of this program is found on the floppy disk itself, and is found in the upper-right corner of the DEV_PC main menu.

**inst_pc**

This file contains the HP-IB addresses for the equipment connected to the HP-IB card. These addresses are set from the DEV_PC main menu when **`Configure System`** and then **`Set System Instrument Address`** is selected.

**mass_pc**

This file contains the mass-storage locations used by DEV_PC to find the main directory path and the project-storage directory path (where your programming work is kept). The mass-storage locations are set up from the DEV_PC main menu when **`Configure System`** and then **`Set Mass Storage Locations`** is selected.

**color_pc**

This file contains the color settings for text and background, used by DEV_PC. The color settings are set from the DEV_PC main menu when **`Configure System`** and then **`Set Text and Background Colors`** is selected.

**xfer.pgm**

This file contains the program that is downloaded into the Test Set in order to transfer library and procedure files from the PC to the test set's memory card or visa versa.

# The Tests Subsystem

The following information illustrates how the code, library, and procedure files are used within the Tests Subsystem (i.e., the `TESTS (Main Menu)` screen).

***NOTE:*** In the chapter titled *Operational Overview of DEV_PC*, you'll find detailed information on using the Tests Subsystem as well as descriptions of the code, library, and procedure files. Also, refer to the *HP 8920 and HP 8921 Programmer's Guide* for specific information about memory card filing systems and how they work with the IBASIC Tests Subsystem.

## File Naming Conventions and File Types

Table 3 shows the default file naming conventions that are used and their associated file types for LIF and DOS formats. Following this table, you will see where the files are used in the `TESTS (Main Menu)` screen.

**Table 3 File Naming Conventions and File Types**

| File Naming Conventions with File Types in () brackets | | | |
|---|---|---|---|
| Test Set Model | HP 8920 or HP 8921 | HP 8920B | File is used in the TESTS (Main Menu) in the following fields: |
| Default File Format | LIF | DOS | |
| Code File (File Type) | `cNAME` (ASCII) | `NAME.PGM` (IBPRG) | Not shown. But located by Code_loc$ in the procedure file |
| Library File (File Type) | `lNAME` (BDAT) | `NAME.LIB` (BDAT) | Shown in the "Library" field, and located by "Lib_name$" in the procedure file |
| Procedure File (File Type) | `pNAME` (BDAT) | `NAME.PRC` (BDAT) | Shown in "Procedure" or "Select Procedure Filename" field and located by "Lib_name$" in the procedure file |

## Library File Description

The contents of a library file cannot be edited except by using DEV_PC. The data structure of a library file has the following elements, in order:

- ❒ Lib_id (library identification; for example,” 8920”)
- ❒ Lib_des$ (library description; for example, “These tests are derived from the EIA standards”)
- ❒ Lib_revid$ (library revision date; for example, “15 DEC 95”)
- ❒ Test$ (test names, up to 50 names; for example, “TX frequency error”)
- ❒ Parm$ (parameter names, up to 50 names; for example, “RT low supply voltage”)
- ❒ Parm (parameter factory-default values, up to 50 values)
- ❒ Parm_units$ (parameter units, up to 50 units)
- ❒ Spec$ (specification names, up to 50 specifications; for example, “TX current drain”)
- ❒ Spec (specification factory-default limits, up to 50 limit pairs)
- ❒ Spec_units (specification units, up to 50 units)
- ❒ Ext_lib (extended libraries are valid or not valid)
- ❒ Freq_c_cntl$
- ❒ Freq_c_title$ (titles in the Edit Frequencies or Channel Information screen)
- ❒ Lib_title$ (title and revision number of the library; a title does not appear in test sets with firmware older than A.14.00 or B.00.00)
- ❒ C_name$ (valid calling names of external equipment used in the Edit Configuration or External Devices screen, up to 10 names may be allowed)
- ❒ C_model$ (valid model numbers of any external equipment defined in C_name$, that may be used with test set, up to 10 may be allowed)
- ❒ C_opt$ (valid entries for the Options field in the Edit Configuration or External Devices screen, up to 10 may be allowed for each piece of external equipment)

## Procedure File Architecture

The contents of a procedure file cannot be edited except by using DEV_PC. The data structure of a procedure file has the following elements, in order:

❒ Proc_id (procedure identification; for example, 8920)
❒ Proc_des$ (procedure description found in the in the TESTS (Main Menu) screen; for example, “This program performs automated tests for FM radios.”)
❒ Lib_name$ (library and procedure name; for example,” NA_FM”, library and procedure name appears in the TESTS (Main Menu) screen)
❒ Code_loc$ (location where program code is to be found; for example, CARD)
❒ Lib_revid$ (library revision identification date; for example, “15 DEC 95”)
❒ Present_tests (the test numbers for the factory-default selected tests displayed in the Edit Sequence or Order of Tests screen, up to 50 tests are allowed)
❒ Parm (parameter factory-default values, up to 50 values are allowed)
❒ Spec (specification factory-default limits, up to 50 upper and lower limits are allowed)
❒ Spec_check (specification check for upper, lower, both or none limits, up to 50 values between 0 and 4 may be entered)
❒ Rx_freq (receive frequency value in MHz, up to 50 frequencies are allowed)
❒ Tx_freq (transmitter frequency value in MHz, up to 50 frequencies are allowed)
❒ Rx_info$ (information entered in the field below the RX frequency; up to 50 information entries are allowed)
❒ Tx_info$ (information entered in the field below the TX frequency; up to 50 information entries are allowed)
❒ Test_it$ (the Y or N answer to the “Test?” question/prompt in the Edit Frequencies or Channel Information screen)
❒ Prime$ (the Y or N answer to the “Prime?” question/prompt in the Edit Frequencies or Channel Information screen)
❒ Pass_number (the password used when a test procedure is saved)
❒ Secure_it (the field determining if the specification and parameters are secured for unauthorized use)

## Extended Library File Architecture

If your Test Set firmware is at revision A.14.00 or B.00.00 or newer, the library and procedure files may take advantage of extended libraries. There are three advantages to extended libraries:

1. The library file title is displayed in a large font at the top of the **`TESTS (Main Menu)`** screen; also, the software revision number is shown in regular font.
2. You are able to customize the titles at the top of the **`Channel Information`** screen.
3. You are able to create “calling names” (which is the name of an external device) and options used in the **`External Devices`** screen.

# DEV_PC Functional Descriptions

## Configure System

This function allows the user to modify a number of default settings that are then accessed whenever DEV_PC is run. These settings are stored in the DEV_PC directory.

### Set Mass Storage Locations

Enter the drive and directory path for the DEV_PC files and project files.

### Set System Instrument Address

Sets the HP-IB address for the test set and printer. The printer can either be on the HP-IB bus, or a local serial or parallel printer, or a network printer.

### Set Text and Background Colors

If the PC monitor is a color, bit-mapped display whose driver supports color mapping, then the text and background colors can be customized.

## Load Files From PC Disk

This selection displays a list of all the procedure files stored in the project directory-path designated under `Configure System`. The user enters a file name and then the data for the specified procedure file is loaded into the DEV_PC program variables. The library file that is associated with the procedure is also loaded into the program variables.

## Define/Mod Data for Files

This selection is used for modification or creation of library and procedure files. If a procedure and library have been previously loaded into DEV_PC program variables, the variables can be accessed through the DEV_PC main menu. The note “(Library File)” or “(Procedure File)” is appended to the following headings as a note showing you which file is affected by that function.

*NOTE:* The Clear data softkey available on this menu sets all library and procedure settings back to the original DEV_PC default values.

### Define Type of TESTS Screen (Library File)

There are two types of TESTS screens. Select “Rev 1” if your test set has firmware below A.14.00. Test sets with firmware A.14.00 or higher (including B.00.00) support both “Rev 1” or “Rev 2” choices. “Rev 2” includes the extended library features and should be used if you are programming an HP 8920B test set.

### Define Library Type, Name and Rev (Library File)

If the code being developed does not require use of a library file, select **`NO LIBRARY`**; however, if you are going to take advantage of the Tests Subsystem features, select **`USER LIBRARY`**. Selecting **`NO LIBRARY`** removes all of the functions under **`Define/Mod Data For Files`** except for two which allow you to enter a procedure name and comments. Selecting **`USER LIBRARY`** allows you to choose the library file that DEV_PC will load or create (if it’s a new library file).

### Define TESTS Header Information (Library File)

This function applies to “Rev 2” TESTS screens only! Enter up to 30 characters of text to appears at the top of the **`TESTS (Main Menu)`** screen in large-font typeface. The second line of header information will show the software’s revision number in normal font size. The revision number should be the same as the revision number shown in the first line of code.

### Define Library Comments (Library File)

Up to 5 lines of comments may be entered. The comments are not accessible from the TESTS screen, but are used solely for the programmer to enter comments pertinent to the library file being developed.

### Define Instruments Supported (Library File)

This function applies to “Rev 2” TESTS screens only! Enter the calling name, model, and options to be displayed in the **`TESTS (External Devices)`** menu.

### Define Frequency Screen Header (Library File)

This function applies to “Rev 2” TESTS screens only! This function allows the customizing of the **`TESTS (Channel Information)`** menu. Two lines of text are allowed to be displayed at the top of the screen. Six fields are user-selectable (can be turned on or off) and definable (can be re-written to display what you want to appear on the test set). Keep in mind that the first two fields require numeric data to be entered, the next two fields expect a Yes or No response, and the last two fields expect string data to be entered by the test-set user.

### Define Specification Default Names (Library File)

Enter up to 50 specifications and the associated units of measure for each specification.

### Define Parameter Default Names (Library File)

Enter up to 50 parameters and the associated units for each parameter.

### Define Test Default Names (Library File)

Enter up to 50 test names. Notice the IBASIC subroutine number for calling any of the tests is shown in the left column of this screen.

### Define Procedure Name (Procedure File)

Enter the test procedure’s name. A listing of existing of existing procedure files is displayed from the specified project-directory path if you need to choose an existing file.

### Define Procedure Comments (Procedure File)

Enter up to 2 lines of text to be displayed on the **`TESTS (Main Menu)`** under the **`Comment`** or **`Description`** area (depending upon the firmware revision in your test set).

### Edit Frequency Values (Procedure File)

Enter receive and transmit frequencies and channel information to be used as the default values for the test procedure. Also, enter whether the TX/RX frequency pair is to be used during testing, and if it is to be a prime channel (that is, will it be used by the tests selected for “All Chans?”). Values you enter are displayed on the test set in the **`TESTS (Channel Information)`** or **`TESTS (Edit Frequencies)`** menu (depending on the firmware revision in your test set).

### Edit Specification Values (Procedure File)

Enter specification values for the lower and upper limits to be used as the default values for the test procedure. Values you enter are displayed on the test set in the **`TESTS (Pass/Fail Limits)`** or **`TESTS (Edit Specifications)`** menu (depending on the firmware revision in your test set).

### Edit Parameter Values (Procedure File)

Enter parameter values to be used as the default values for the test procedure. Values you enter are displayed on the test set in the **`TESTS (Test Parameters)`** or **`TESTS (Edit Parameters)`** menu (depending on the firmware revision in your test set).

### Edit Test Sequence (Procedure File)

Enter tests from the list of Test Default Names previously defined; the tests you select are to be used as the default tests displayed in the **`TESTS (Order of Tests)`** or **`TESTS (Edit Sequence)`** menu (depending on the firmware revision in your test set).

**Secure Test Information (Procedure File)**

Selecting this function allows you to secure certain menus from being accessed by the test-set user. The following menus can be secured:

- `TESTS (Order of Tests)` or `(Edit Sequence)`
- `TESTS (Channel Information)` or `(Edit Frequencies)`
- `TESTS (Pass/Fail Limits)` or `(Edit Specifications)`
- `TESTS (Test Parameters)` or `(Edit Parameters)`
- `TESTS (External Devices)` or `(Edit Configuration)`

## Get Files From Card Media

This function loads both the library and procedure files into DEV_PC.

## Put Files On Card Media

This function puts the library and procedure files currently residing in DEV_PC, into the test set's memory card. The program "xfer.pgm" is used to transfer the library and procedure files to the test set. This program is first loaded into the test set's internal RAM memory; any code currently residing in RAM memory is deleted.

## Enter Test Set Commands

This function allows you to directly enterIBASIC commands to the test set over the HP-IB bus.

## Get Code From Test Set

This function causes the IBASIC code file currently in the test set's RAM memory to be uploaded into the PC and saved under the specified project-directory path.

## Put Code on Card Media

This function allows you to select a code file from the specified project-directory path to be put onto the test set's memory card. To achieve this, the code is first transferred into the test set's RAM memory.

## Edit/Create a Code File

This function allows you to develop the code file. The DEV_PC program is automatically exited and you are prompted to enter the code file name to be used. When editing code you may use the HP-IB address to execute the file directly from the PC; however, before downloading the code into the test set's RAM memory using the F1 through F4 function keys, change the HP-IB address to 800 in order for the code to work on the internal HP-IB bus in the test set.

## Print Test Set Data

This function prints the library and procedure file data currently loaded in DEV_PC to the printer specified under the `Configure System` function.

## Program Structure for Tests Subsystem

Writing programs that take advantage of the Tests Subsystem capabilities requires the programmer to understand how to structure the program to access the Tests Subsystem user-interface screens.

### General Organization

The following steps show a basic algorithm that can be used to execute a number of test subroutines at a number of different test frequencies:

```
BEGIN
SET UP (Set up the COM area to hold the global variables.)
REPEAT (for all Test Frequencies)
    REPEAT (for all Defined Tests)
        DO SUBROUTINE (defined Test)
    UNTIL (All Defined Tests Done)
UNTIL (All Test Frequencies Tested)
END
SUBROUTINE1 (Defined Test 1)
SUBROUTINE2 (Defined Test 2)
SUBROUTINE3 (Defined Test 3)
```

## Program Example

The following IBASIC program is an example that uses the basic algorithm shown on the previous page and the Tests Subsystem to execute a number of test subroutines at a number of defined test frequencies. Also included are examples of how to interact with the user-interface to allow a user to access parameters, specifications, and configuration fields to define a specific set of test requirements.

An explanation of the program example is given at the end of the listing.

### Program Listing

```
10    ! DEMO_1 Rev A.00.00
20    !
30    ! THE FIRST LINE MUST CONTAIN THE NAME OF THE LIBRARY
40    !
50    !__________________________________________________________
60    !
70    ! THIS PROGRAM IS A DEMO PROGRAM TO DEMONSTRATE THE USE
80    ! OF THE TEST SUBSYSTEM ON THE Test Set
90    !
100   ! REVISION: 1 APRIL, 1991
110   !___________________________________________________________
120   !
130   COM /I_o/ I_o$[470]
140   ! INPUT OUTPUT STRING
150   COM /Freq/ Rx_f,Tx_f
160   ! PRESENT RX AND TX FREQUENCIES IN MHZ
170   !
180   INTEGER Test_return
190   !  TITLE SCREEN FOR OUR TESTS
200   CLEAR SCREEN
210   PRINT TABXY(2,2),"___DEMO PROGRAM FOR THE TESTS SUBSYSTEM___"
220   !
230   !  SET UP A SOFT KEY TO HALT THE PROGRAM
240   ON KEY 1 LABEL "Stop Test",5 GOTO Stp_test
250   !
260   !  CLEAR THE INTERNAL Test Set BUS
270   CLEAR 800
280   !

290   ! NOW READ THE TEST FREQUENCIES IN ONE AT A TIME AND DO THE
300   ! SEQUENCE OF TESTS ON THEM
310   Ch=1
320   REPEAT
330     OUTPUT 800;"TESTS:FREQ? "&VAL$(Ch)
340     I_o$=""
350     ENTER 800;I_o$
360     !  SET THE VALUE OF THE RX FREQUENCY
370     Rx_f=VAL(I_o$[4;12])
380     !  SET THE VALUE OF THE TX FREQUENCY
390     Tx_f=VAL(I_o$[30;12])
400     !  SET WHETHER TO TEST THIS FREQUENCY
410     T_it$=I_o$[56;1]
420     !  SET IF THIS IS A PRIME FREQUENCY
430     IF (LEN(I_o$)>57) THEN
440       Prime$=I_o$[58;1]
450     ELSE
460       Prime$="N"
```

```
470     END IF
480     !  IF THIS FREQUENCY IS TO BE TESTED
490     IF T_it$="Y" THEN
500       PRINT TABXY(2,6),"RX FREQUENCY = ",Rx_f
510       PRINT TABXY(2,7),"TX FREQUENCY = ",Tx_f
520       PRINT TABXY(2,8),"TEST THIS FREQUENCY ?",T_it$
530       Run_ts=1
540       !  RUN THROUGH THE SEQUENCE OF TESTS
550       REPEAT
560         Done_t=0
570         !  ENTER IN THE TEST SEQUENCE
580         OUTPUT 800;"TESTS:SEQN? "&VAL$(Run_ts)
590         I_o$=""
600         ENTER 800;I_o$
610         Tst=VAL(I_o$[4;2])
620         !  IF THIS TEST IS TO BE SKIPPED THEN SET THIS
630         IF I_o$[7;1]="N" THEN Tst=-Tst
640         !  IF THIS IS A PRIME FREQUENCY RUN THE TEST
650         IF Tst<0 AND Prime$="Y" THEN
660         !  CALLS THE SUBROUTINE NAME T(ABS(Tst))
670           T(ABS(Tst),Test_return)
680           IF (Test_return=1) THEN GOTO Test_error
690           Done_t=1
700         END IF

710         !  IF THIS TEST IS TO BE DONE AND IS NOT A PRIME FREQUENCY
720         IF Tst>0 AND NOT Done_t THEN
730         !  CALLS THE SUBROUTINE NAME T(ABS(Tst))
740           T(ABS(Tst),Test_return)
750           IF (Test_return=1) THEN GOTO Test_error
760         END IF
770         Run_ts=Run_ts+1
780       UNTIL Tst=0 OR Run_ts=51
790     END IF
800     Ch=Ch+1
810   UNTIL Ch=51 OR Tx_f=-1 OR Rx_f=-1
820 Stp_test:  !
830   CLEAR SCREEN
840   PRINT TABXY(2,10),"FINISHED TESTING"
850   GOTO End_program
860 Test_error:  !
870   CLEAR SCREEN
880   PRINT TABXY(2,10),"PROGRAM STOPPED, TEST ",ABS(Tst),"FAILED"
890 End_program:  !
900   END
910 T01:SUB T01(Test_return)
920     COM /I_o/ I_o$
930     COM /Freq/ Rx_f,Tx_f
940     DIM Calling_name$[22],Model$[22],Options$[22]
950   !  TEST ROUTINE NUMBER 1
960     PRINT TABXY(2,12),"DOING TEST NUMBER 1 FOR FREQ ",Rx_f
970     !  GET THE PARAMETER 1 FOR THIS TEST
980     OUTPUT 800;"TESTS:PARM? "&VAL$(1)
990     I_o$=""
1000    ENTER 800;I_o$
1010    !  IF THERE IS NO PARAMETER THEN PAUSE
1020    IF I_o$[1;5]="Error" THEN
1030      PRINT TABXY(2,14),"ERROR IN RECALLING THE PARAMETERS FOR TEST 1"
1040      Test_return=1
1050    END IF
```

```
1060    Parm_1=VAL(I_o$)
1070    !  GET CONFIGURATION 1 INFO FOR THIS TEST
1080    OUTPUT 800;"TESTS:CONF? "&VAL$(1)
1090    I_o$=""
1100    ENTER 800;I_o$
1110    Calling_name$=I_o$[4;21]
1120    Model$=I_o$[27;21]
1130    I1addr=VAL(TRIM$(I_o$[50]))
1140    Options$=I_o$[54]
1150    !  GET SPECIFICATION 1 FOR THIS TEST
1160    OUTPUT 800;"TESTS:SPEC? "&VAL$(1)
1170    I_o$=""
1180    ENTER 800;I_o$
1190    IF I_o$[1;5]="Error" THEN
1200      PRINT TABXY(2,14),"ERROR IN RECALLING THE SPECIFICATIONS FOR TEST 1"
1210      Test_return=1
1220    END IF
1230    Lower_limit=VAL(TRIM$(I_o$[4]))
1240    Upper_limit=VAL(TRIM$(I_o$[17]))
1250    Test$=TRIM$(I_o$[30])
1260  SUBEND
1270 T02:SUB T02(Test_return)
1280    COM /I_o/ I_o$
1290    COM /Freq/ Rx_f,Tx_f
1300  !  TEST ROUTINE NUMBER 2
1310    PRINT TABXY(2,13),"DOING TEST NUMBER 2 FOR FREQ ",Rx_f
1320  SUBEND
1330 T03:SUB T03(Test_return)
1340    COM /I_o/ I_o$
1350    COM /Freq/ Rx_f,Tx_f
1360  !  TEST ROUTINE NUMBER 3
1370    PRINT TABXY(2,14),"DOING TEST NUMBER 3 FOR FREQ ",Rx_f
1380  SUBEND

1390 T:SUB T(N,Test_return)
1400    ! CALL THE PASSED TEST NUMBER (N)
1410    SELECT N
1420    CASE 1
1430      T01(Test_return)
1440    CASE 2
1450      T02(Test_return)
1460    CASE 3
1470      T03(Test_return)
           ""
           ""
           ""
2380    CASE 49
2390      T49(Test_return)
2400    CASE 50
2410      T50(Test_return)
2420    END SELECT
2430  SUBEND
```

### Program Listing Explanation

**10** This first line must contain the name of the library and the program. This is checked by the Tests Subsystem when loading the program. Also, the revision number for the code must be here, otherwise the code is loaded into the Test Set every time the Run Test key is pressed.

**130** Establish a common I_o$ string for the ENTER statements.

**150** Establish common Rx_f and Tx_f that can be used by the subprograms (tests).

**180** The Integer Test_return is used by the subprograms to indicate the test ended with some error condition. The meaning of Test_return could be expanded to include the status of the test (for example, PASS/FAIL).

**200** Clears the IBASIC screen.

**210** Prints and indication that the Demo program is running.

**240** Allows the User to stop the program using a softkey.

**270** Clear the Internal Bus of the Test Set

**310** Ch keeps track of which channel we are currently testing.

**320** Now Repeat for all Frequencies

**330** Request all the channel values from the Test Set.

**340** I_o$ gets the string return.

**370** The Rx frequency is pulled from the string.

**390** The Tx frequency is pulled from the string.

**410** T_it$ gets either a “Y” or an “N” depending on whether this frequency is to be tested.

**430** If a Prime channel has been specified then Prime$ gets a value of “Y”.

**490** If this frequency is to be tested

**500-520** Print out some information on the test about to be performed.

**530** Run_ts holds the value of the test currently being run.

**550** Repeat for all Specified Tests

**560** Done_t is initialized to not completed.

**580** Get the Test specifier for the current Test.

**590** Initialize I_o$ to a null string.

**600** I_o$ holds the value of the return string.

**610** Tst now hold the value of the current test. This value is equal to the index of the Test Name in the Test selection list shown on the Test Seqn screen.

**630** This tests whether this test is to be run for all channels. If not, the value is still kept around but is made negative. This will be used in later tests.

**650** If the number of the test is indeed negative but the channel is prime, then the test is done.

**670** This calls a subroutine that maps the number of the test with the subroutine that defines this test.

**680** If there is an error, then the program stops and the error is reported.

**690** Done_t is set to completed.

**700** End this IF statement.

**720** If Tst is suppose to be done, and has not yet been done, then now do it.

**740** Again, This calls a subroutine that maps the number of the test with the subroutine that defines this test.

**750** If there is an error, then the program stops and the error is reported.

**760** End this IF statement.

**770** Increment the step for the Test index.

**780** If there are no more steps specified, or if the number of tests run is 51, then leave the test seqn loop.

**790** End the Tst IF statement.

**800** Increment the Channel number.

**810** Stop stepping through the channels if the number of channels reaches 51, or if the Receive or Transmit frequencies are specified at –1.

**820** The goto location for the stop test softkey.

**830** Clear the screen

**840** Indicate that the test is finished.

**850** Goto the end statement.

**860** The goto location if an error occurs in one of the subroutines.

**870** Clear the screen.

**880** Indicate that one of the tests have failed.

**890** The goto for the end of the program.

**900** End of the main program.

**910** Subroutine T01-This corresponds with test #1. This subroutine illustrates how to enter values from the Parameters, Configuration, and Specification screens.

**920-930** Includes the common variables.

**940** Dimension some variables that will be used to store values from the configure screen.

**960** Indicate that the first test is now active.

**980** Enter the value of the first Parameter. This is the value of the first parameter on the Parameter screen.

**990** Initialize the I_o$ string.

**1000** Enter the value.

**1020-1050** If there is no defined parameter this string will catch the error and return it to the main program.

**1080** Get the information for the first instrument stored on the configure screen.

**1090** Initialize the I_o$ string.

**1100** Enter the string.

**1110** Calling_name$ now holds the string associated with the Calling Name field on the configure screen.

**1120** Model$ now holds the string associated with the Model field on the configure screen.

**1130** I1addr equals the value in the Addr field on the configure screen.

**1140** Options$ now holds the string associated with the Options field on the configure screen.

**1160** Get the information for the first Specification listed on the Specification system.

**1170** Initialize the I_o$ string to null.

**1180** ENTER the I_o$ string.

**1190-1220** If there is no specification defined for this specification number, then an Error will appear in the I_o$ string. If this occurs, stop the test and return the error to the main program.

**1230** Set the lower limit from the value in the string.

**1240** Set the upper limit from the value in the string.

**1250** Set Test$ to whether “Upper”, “Lower”, “Both”, or “None” of the specs are to be tested.

**1260** End of this subroutine.

**1270-1380** These are the second and third subroutines. They are labeled T02 and T03 to correspond with the second and third test routines defined on the Test Seqn screen.

**1390-2430** SUB T maps the calls from the main program to the correct subroutine. The mapping is quite simple, with the main program specifying which test to run and this subroutine calling the correct subroutine based on the SELECT statement.

### Creating A Library And Default Procedure File

Once the code file has been created, an associated library and default procedure file for the code file can also be created. Both of these files are BDAT files and are created using the DEV_PC program.

### Creating A Procedure File With No Library

If the programmer does not want the program to use the different user-interface screens of the Tests Subsystem, create a procedure from the code file that does not have a library associated with it. This is done using the DEV_PC program. When the test information is defined, [NO LIB] is selected for the library name.

When creating a procedure to run without a Library, the base name of the code file must be the same as the base name of the procedure file. Also, the first line of the code file must be an exclamation point followed by the procedure file name. For example, if the procedure is called FM_TESTS the first line of your code file **must** be

```
1    ! FM_TESTS
```

# Index

# Index

# Index

# Index

# Index