VIEWDAC
Getting Started

KEITHLEY

53207 Rev. A

New Contact Information

Keithley Instruments, Inc.
28775 Aurora Road
Cleveland, OH 44139

Technical Support: 1-888-KEITHLEY
Monday – Friday 8:00 a.m. to 5:00 p.m (EST)
Fax: (440) 248-6168

Visit our website at http://www.keithley.com

The information contained in this manual is believed to be accurate and reliable. However, Keithley Instruments, Inc., assumes no responsibility for its use or for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of Keithley Instruments, Inc.

KEITHLEY INSTRUMENTS, INC., SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATED TO THE USE OF THIS PRODUCT. THIS PRODUCT IS NOT DESIGNED WITH COMPONENTS OF A LEVEL OF RELIABILITY SUITED FOR USE IN LIFE SUPPORT OR CRITICAL APPLICATIONS.

Refer to your Keithley Instruments license agreement and Conditions of Sale document for specific warranty and liability information.

Keithley MetraByte Division

Keithley Instruments, Inc.

440 Myles Standish Blvd. Taunton, MA 02780

Telephone: (508) 880-3000 • FAX: (508) 880-0179

Revision A - April 1994

Part Number: 53207

Preface

VIEWDAC - Getting Started describes how to install and run VIEWDAC in your computer, provides an overview of the navigational tools used to get around in the VIEWDAC environment, and provides an overview of VIEWDAC features. It also describes how to use the VIEWRUN run-time system.

The manual is intended for system designers who are using VIEWDAC to create data acquisition application programs.

It is assumed that you are familiar with data acquisition principles, with your particular application, and with the computer you are using. It is also assumed that you are familiar with standard windowing-type environments, including the use of a mouse, a keyboard, pull-down and pop-up menus, dialog boxes, and the clipboard. If you are not familiar with a windowing-type environment, it is recommended that you look in your local library or computer bookstore for a book that provides an overview of these concepts. One suggested title is *At Home with Windows* by Paul McFedries (Alpha Books, A Division of Prentice Hall Computer Publishing, 11711 North College, Carmel, IN 46032).

Overview of Contents

VIEWDAC - Getting Started is organized as follows:

- **Chapter 1: Installation and Startup**, describes how to install and run VIEWDAC in your computer.

- **Chapter 2: Getting Around in VIEWDAC**, provides an overview of the navigational tools used to get around in the VIEWDAC environment.

- **Chapter 3: VIEWDAC at a Glance**, provides an overview of VIEWDAC features.

- **Chapter 4: Troubleshooting**, provides information about how to solve common problems you may encounter when using VIEWDAC.

- **Appendix A: Memory Usage in VIEWDAC**, provides some general information about the way memory is used in VIEWDAC.

- **Appendix B: Summary of VIEWDAC Menus**, provides a summary of all VIEWDAC menus and the options available with each.

- **Appendix C: Operators and Functions**, provides a summary of mathematical and logical operators and mathematical functions used in VIEWDAC.

- **Appendix D: Using VIEWRUN**, describes how to use the VIEWRUN run-time system.

An index completes this manual.

Note: *VIEWDAC - Getting Started* replaces the following documents: *VIEWDAC Roadmap*, *VIEWDAC Environment*, *Encore - a quick reference to VIEWDAC*, and *VIEWRUN - The VIEWDAC Run-Time System*.

Documentation Conventions

Keep the following conventions in mind as you use this manual:

- Information that you must type is shown in `courier` typeface. For example, to load the demonstration driver, enter `DASDEMO` at the DOS prompt. Unless otherwise specified, you can type characters in either UPPERCASE or lowercase.

- Variable information is shown in *italics*. For example, *filename.ext* denotes that you should supply an appropriate file name and extension.

- Keyboard keys are denoted by square brackets surrounding the key's label. For example, [Esc] is the escape key, [Del] is the delete key, [F2] is the F2 function key, and so on.

- If two keys are joined by a plus symbol, you should hold down the first key while you press the second key. For example, [Ctrl]+[Break] denotes that you should hold down [Ctrl] while you press [Break].

- If two keys are separated by a space, you should press the keys sequentially. For example, [F] [spacebar] denotes that you should press and release [F] and then press and release [spacebar].

- A specific arrow key is denoted by the direction in which its arrow points: [up arrow], [down arrow], [right arrow], and [left arrow].

- The term "enter" indicates that you should press the specified key or type the specified characters and then press [Enter] or [Return].

- The terms "press" and "type" indicate that you should press the specified key or type the specified characters without pressing [Enter].

- The term "click" indicates that you should either highlight an item or initiate an action using the mouse. For example, to click on the OK push button, move the mouse pointer to the OK push button and then press the left mouse button.

- The term "select" indicates that you should either highlight an item or initiate an action using either the mouse or the keyboard. For example, you can select the OK push button in a dialog box either by clicking on OK or by using [Tab] to move to the OK push button and then pressing [spacebar].

- The term "menu" refers to pull-down menus. Pop-up menus are always referred to as pop-up menus.

- The forward slash (/) indicates an either/or selection. For example, ON/OFF means the result is either on or off, but not both.

- The back slash (\) denotes a sequence of menu selections. For example, File\New\Text Edit denotes that you should first select the File option from the menu bar, then select the New option from the File menu, and then select the Text Edit option from the New menu.

Related Documentation

Refer to the following documentation for more information about using VIEWDAC:

- **VIEWDAC Tutorials -** This manual provides tutorials for using VIEWDAC. Read this manual when you feel comfortable with VIEWDAC's visual interface and are ready to begin learning to use VIEWDAC's sequences. The first two chapters introduce sequences, which are groups of tasks or actions; the remaining chapters provide specialized information, including graphics, data processing, interactive data analysis, curve fitting, and using macros.

- **VIEWDAC Reference -** This manual includes definitions of VIEWDAC concepts, descriptions of all features prior to VIEWDAC 2.2, a glossary, and a comprehensive index. Use the individual task category chapters to decide which tasks to choose and how to set them up.

- **VIEWDAC - What's New in Version 2.2 -** This manual contains information about the new features in VIEWDAC 2.2. It also contains information that was either missing from or incorrect in the *VIEWDAC Reference* manual.

- **Read Me First brochure -** This brochure contains warranty information, a registration card, customer support information, and the package serial number.

- **Documentation on disk -** The following files are provided on disk:
  - **README.1ST -** Includes last-minute VIEWDAC information.
  - **DEVICES.DOC -** Contains a complete list of supported graphics adapters.
  - **DEMOS.DOC -** Describes the demonstration sequences provided with VIEWDAC.
  - **BDE.DOC -** Contains additional printer configuration information.
  - **FILES.DOC -** Contains a brief description of all files provided with the VIEWDAC software package.

Table of Contents

Preface

1 Installation and Startup

2 Getting Around in VIEWDAC

3 VIEWDAC at a Glance

List of Figures

List of Tables

1

Installation and Startup

VIEWDAC is a data acquisition software package that allows you to develop data acquisition and control, analysis, and graphics application programs (called sequences in VIEWDAC). You can create sequences for a wide variety of analog input/output (I/O), digital I/O, General Purpose Interface Bus (GPIB), and RS-232 devices. You can also use VIEWDAC to create custom-designed display panels that allow you to monitor and to alter your sequences.

VIEWDAC runs under DOS, using a graphical, windowing-type interface. Generally, the way you use menus, windows, dialog boxes, the mouse, the keyboard, and the clipboard in VIEWDAC is the same as the way you would use these features in any standard windowing-type environment. Refer to Chapter 2 for information about the specific way these features are used in VIEWDAC.

VIEWRUN is a run-time version of VIEWDAC. VIEWRUN allows users to run, but not modify, sequences that you create with VIEWDAC. Refer to Appendix D for more information about VIEWRUN.

This chapter contains the information you need to install and run VIEWDAC and to create configuration files.

System Requirements

To run VIEWDAC, your system must consist of the following:

- IBM PC or compatible computer, with a 386DX, 386SX, 486DX, 486SX, or Pentium microprocessor.

- MS-DOS® (Version 3.1 or higher)

- At least 6M bytes of Random Access Memory (RAM), typically configured as follows:

  – 640K bytes of conventional memory

  – 384K bytes of high memory

  – 5M bytes of extended memory

  Refer to Appendix A for information about how memory is used in VIEWDAC.

- Hard disk drive with at least 15M bytes free.

- EGA-compatible or VGA-compatible graphics adapter. (Note that many graphics adapters with greater resolution than VGA are also supported.)

- Mouse / Trackball with Microsoft®-compatible driver. (To perform all VIEWDAC functions, the mouse must have at least two buttons.)

- One of the following math coprocessors:

  – 80387 for the 386DX

  – 80387SX for the 386SX

  – 80487SX for the 486SX

  Note that no math coprocessor is needed for 486DX and Pentium machines.

- For data acquisition operations, a Keithley DAS (data acquisition system) board and appropriate external DAS driver. Contact Keithley MetraByte or your local sales representative for a list of supported boards.

- (Optional) An expanded memory manager, such as QEMM/386, 386MAX, or EMM386. Refer to Appendix A for information about using expanded memory in VIEWDAC. If you use an expanded memory manager, it must be VCPI (Virtual Control Program Interface)-compliant.

Running VIEWDAC

To run VIEWDAC, perform the quick installation procedure, as described in the following subsection. If VIEWDAC does not run exactly as you want it to, set up a customized environment for running VIEWDAC; refer to page 1-5 for more information.

Setting up the VIEWDAC environment may require the following:

- Loading an external DAS driver. Refer to page 1-8 for more information.
- Installing the hardware protection key. Refer to page 1-9 for more information.
- Loading a mouse driver. Refer to page 1-10 for more information.
- Setting the display mode. Refer to page 1-10 for more information.
- Setting the memory mode. Refer to page 1-11 for more information.

Quick Installation and Startup

To perform a quick installation and startup of VIEWDAC, perform the following steps:

Note: It is recommended that you make a back-up copy of your VIEWDAC disks before you begin.

1. Insert Disk 1 in the appropriate disk drive.

2. Run the VIEWDAC installation program. For example, assuming that Disk 1 is in disk drive A, enter the following at the DOS prompt:

   ```
   A:install
   ```

The installation program prompts you for your installation preferences, including the name of the directory you want to copy the software to. It also prompts you to insert additional disks, as necessary.

Caution: If you are upgrading from a previous version of VIEWDAC and have sufficient hard disk space, it is recommended that you maintain both the new and old versions of VIEWDAC initially. If you cannot maintain both versions, you must convert binary sequences created in the old version to ASCII sequences before you install the new version. Refer to *VIEWDAC - What's New in Version 2.2* for more information. If you can maintain both versions, either rename the directory containing the old version of VIEWDAC before you install the new version, or specify a different directory for the new version at the prompt.

---

3. Continue to insert disks and respond to prompts, as appropriate.

   The installation program decompresses the VIEWDAC files, deletes any temporary files, and then copies the files into the directory you specified (C:\VIEWDAC if you do not specify otherwise). Note that the installation program does not modify your AUTOEXEC.BAT file or your CONFIG.SYS file.

4. Making sure that you are still in the directory containing the VIEWDAC software, start VIEWDAC by entering the following at the DOS prompt:

   ```
   VIEWDAC
   ```

   The VIEWDAC startup screen is displayed.

Note: Normally, you cannot run VIEWDAC from the MS-DOS Prompt when in the Microsoft Windows™ operating environment. If Windows is running when you start VIEWDAC, the system displays a warning message and the following prompt:

```
Abort the loading of VIEWDAC ([y]/n)?
```

It is recommended that you press [Y] or [Enter] to cancel the operation. Then, exit the Windows environment and try again. If you press [N] to try to start VIEWDAC from Windows, the startup may not be successful.

5. Familiarize yourself with the features of VIEWDAC. Refer to Chapter 2 and Chapter 3 of this manual, the *VIEWDAC Tutorials*, the *VIEWDAC Reference* manual, and *VIEWDAC - What's New in Version 2.2* for more information.

6. To end your VIEWDAC session and return to DOS, select System\Bye from the menu bar.

Setting Up the VIEWDAC Environment

If the default VIEWDAC environment, provided in the quick installation procedure, is not appropriate for your needs, you can customize the VIEWDAC environment by setting environment variables. The VIEWDAC environment variables are described in Table 1-1.

Table 1-1. Environment Variables

Environment Variable	Purpose	Default Setting	Options	More Information
VIEWVIRT	Load an external DAS driver that is not VCPI-compliant	VCPI-compliant external DAS driver	**-intmap 8** indicates an external DAS driver that is not VCPI-compliant	Loading an External DAS Driver, page 1-8
	Specify the memory mode	Virtual memory mode	**-novm** indicates virtual memory mode is disabled	Setting the Memory Mode, page 1-11
	Specify the location of the swap file	Directory containing the VIEWDAC software	**-swd *drive:path***, where *drive:path* is the directory containing the swap file	Setting the Memory Mode, page 1-11
	Specify the size of the swap file	All unused space on hard drive is available	**-maxs *nbytes***, where *nbytes* is the maximum size of the swap file in bytes	Setting the Memory Mode, page 1-11
VIEWDISP	Specify a mouse driver	Mouse driver supplied with your mouse	**/M:1** indicates VIEWDAC mouse driver on COM1 **/M:2** indicates VIEWDAC mouse driver on COM2 **/M:D** indicates mouse driver supplied with your mouse	Loading a Mouse Driver, page 1-10
	Specify the display mode	EGA or VGA graphics adapter	Refer to the DEVICES.DOC file for a list of graphics adapters and their display mode settings.	Setting the Display Mode, page 1-10

Table 1-1. Environment Variables (cont.)

Environment Variable	Purpose	Default Setting	Options	More Information
VIEWCFG	Specify the location of the VIEWDAC or VIEWRUN configuration file	Directory containing the VIEWDAC or VIEWRUN software	***[drive][directory] filename.ext***, where *[drive][directory] filename.ext* is the file containing the configuration information	Saving System Preferences in a Configuration File, page 1-13 Creating a VIEWRUN Configuration File, page D-4
VIEWINI	Specify the location of the VIEWDAC or VIEWRUN initialization file	Directory containing the VIEWDAC or VIEWRUN software	***[drive][directory] filename.ext***, where *[drive][directory] filename.ext* is the file containing the initialization information	Appendix A of the *VIEWDAC Reference* manual Creating a VIEWRUN Initialization File, page D-6

You can set environment variables at the DOS prompt each time you run VIEWDAC, or you can simplify setting up the VIEWDAC environment by including environment variables in a batch file. You can include the environment variables in the following types of batch files:

- **AUTOEXEC.BAT batch file -** Use your AUTOEXEC.BAT batch file for commands that you want to execute every time you start your computer. For example, assume that you want to store configuration information in the same configuration file, specify the same mouse driver, and specify the same location for your swap file every time you start your computer. If you include the VIEWCFG, VIEWDISP, and VIEWVIRT environment variables in your AUTOEXEC.BAT batch file, you will not have to enter these settings every time you start your computer.

- **User-defined batch file -** Use a user-defined batch file for commands that you want to execute only in particular situations. For example, assume that you want to create several VIEWDAC sequences with each one using a different configuration file. You could create several user-defined batch files; in each batch file, you would set the VIEWCFG environment variable to a different configuration file.

Note: Any commands already included in your AUTOEXEC.BAT batch file should not be included in a user-defined batch file.

You can use each environment variable only once. If you are using the same environment variable for multiple purposes, you must include all required options on the same line at the DOS prompt or on the same line in the batch file. For example, if you want to change the file containing the swap file to D:\TEMP, limit the size of the swap file to 10M bytes, and load an external DAS driver that is not VCPI-compliant, you must combine the options, as in the following example:

```
SET VIEWVIRT=-swd D:\TEMP -maxs 10485760 -intmap 8
```

Loading an External DAS Driver

If you are performing data acquisition operations, you should install all Keithley DAS boards and their associated external DAS drivers before you run VIEWDAC. The external DAS drivers allow you to use Keithley DAS boards with VIEWDAC. Refer to the appropriate Keithley DAS board user’s guide for information on installing the board. Refer to the appropriate external DAS driver documentation for information on installing the external DAS driver.

By default, VIEWDAC assumes that your external DAS driver is VCPI-compliant. If you created your own external DAS driver or if you are using an older external DAS driver that is not VCPI-compliant, you must set the VIEWVIRT environment variable as follows:

```
SET VIEWVIRT=-intmap 8
```

If you want to use VIEWDAC without a Keithley DAS board installed, you can use the demonstration driver DASDEMO. To load the demonstration driver, enter the following at the DOS prompt:

```
DASDEMO
```

The DASDEMO driver generates its own data; it remains in memory until you restart the computer.

Installing the Protection Key

If your version of VIEWDAC is shipped with a hardware protection key, you must attach the protection key to your computer before you can run VIEWDAC. To install the protection key, perform the following steps:

1. Select any parallel port on your computer.

   Note: You can use the protection key and another device on the same parallel port. The protection key does not interfere with the performance of the device attached to the parallel port.

2. If necessary, remove any cable that is already attached to the parallel port you want to use.

3. Attach the protection key to the parallel port; you can attach the protection key in only one way.

4. If necessary, reattach the cable to the protection key.

Note: If you are using more than one hardware protection key, it is recommended that you attach VIEWDAC’s protection key closest to the computer. If VIEWDAC’s protection key is not closest to the computer, VIEWDAC may not start up correctly.

Loading a Mouse Driver

VIEWDAC requires a mouse and a mouse driver. You can use either the mouse driver provided by VIEWDAC or the mouse driver supplied with your mouse.

Notes: If you use the mouse driver provided by VIEWDAC, you must use a serial mouse; you can connect the serial mouse to either the COM1 or COM2 port.

If you use the mouse driver supplied with your mouse, make sure that it is Microsoft-compatible.

By default, VIEWDAC assumes that you are using the mouse driver supplied with your mouse. To use VIEWDAC's mouse driver, you must add the /M:*x* option to the VIEWDISP environment variable, where *x* = 1 for COM1 and *x* = 2 for COM2. For example, to use the mouse driver provided with VIEWDAC with a serial mouse on COM2, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWDISP=/M:2
```

You can add the /M:D option to the VIEWDISP environment variable to use the default mouse driver (the mouse driver supplied with your mouse).

If you are not using the mouse driver provided by VIEWDAC, refer to your mouse documentation for instructions on loading the mouse driver into memory.

Setting the Display Mode

If you are using an EGA or VGA graphics adapter, VIEWDAC automatically determines the display mode. If you are using another graphics adapter or if you want to use your EGA or VGA graphics adapter in a different mode, such as monochrome mode, you must use the VIEWDISP environment variable.

The most common display mode settings are listed in Table 1-2. For a complete list of supported graphics adapters and the appropriate settings for each, refer to the DEVICES.DOC file.

Table 1-2. Display Mode Settings

Setting	Graphics Adapter	Resolution	Colors
/I:12	IBM, EGA	640 x 350	16
/I:23	IBM, VGA	640 x 480	16
/I:24	IBM, MCGA-VGA	640 x 480	2
/F:41	IBM, EGA	640 x 350	2

For example, to use a 2-color, IBM EGA graphics adapter, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWDISP=/F:41
```

Setting the Memory Mode

By default, VIEWDAC runs in virtual memory mode. The hard drive provides additional space when no more memory is available. Only the most recently used portions of VIEWDAC are stored in memory. Other portions of VIEWDAC are swapped between memory and the hard drive, as needed.

To provide the additional space, VIEWDAC creates a temporary file, called a swap file, on the hard drive containing the VIEWDAC software. For example, if you installed VIEWDAC in C:\VIEWDAC, VIEWDAC creates the swap file in the VIEWDAC directory on hard drive C. You can change the location of the swap file, if desired, by using the VIEWVIRT environment variable as follows:

```
SET VIEWVIRT=-swd drive:path
```

where *drive:path* is the directory containing the swap file.

For example, to create the swap file in the TEMP directory on hard drive D, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWVIRT=-swd D:\TEMP
```

Note: Do not end the path specification with a backslash (\).

VIEWDAC assumes that all unused space on the hard drive is available for the swap file. You can limit the size of the swap file, if desired, by using the VIEWVIRT environment variable as follows:

```
SET VIEWVIRT=-maxs nbytes
```

where *nbytes* is the maximum size of the swap file in bytes.

For example, to limit the size of the swap file to 10M bytes, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWVIRT=-maxs 10485760
```

Virtual memory mode allows you to work with larger amounts of data; however, VIEWDAC runs more slowly in virtual memory mode. If everything you need to perform your tasks is present in memory, you may not notice a reduction in performance; however, if VIEWDAC must constantly swap between memory and the hard drive, performance is significantly reduced. Note that performance reduction depends on the amount of memory available.

Note: When using virtual memory mode, your hard drive must provide enough space to execute the functions you are trying to perform.

To disable the use of virtual memory mode, you can use the VIEWVIRT environment variable as follows:

```
SET VIEWVIRT=-novm
```

If you disable virtual memory mode, the hard drive is not used. The VIEWDAC software and all data reside in memory.

Saving System Preferences in a Configuration File

VIEWDAC maintains system preferences in a configuration (CFG) file. The CFG file includes the following information:

- System configuration
- Sequence configuration
- GPIB configuration
- GPIB device selection
- RS-232 port configuration
- Hardcopy configuration
- Window preferences
- Snap-to-grid setting
- Macro settings
- **printer.palette** settings
- **day.names** settings
- **month.names** settings

Refer to the *VIEWDAC Reference* manual and to *VIEWDAC - What's New in Version 2.2* for a detailed description of the types of information maintained in the CFG file.

Whenever you change any of this information and select System\Save Options from the menu bar, the CFG file is created (if it does not already exist) or updated (if it does exist). The default CFG file is called VIEWDAC.CFG. By default, when you run VIEWDAC, the software looks for VIEWDAC.CFG in the directory containing the VIEWDAC software. To locate the configuration information in a different file and/or path, you can use the VIEWCFG environment variable as follows:

```
SET VIEWCFG=[drive][directory]filename.ext
```

where *[drive][directory]filename.ext* is the file containing the configuration information.

For example, to locate the configuration information in a CFG file called DEMO.CFG in the CONFIG directory on hard drive D, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWCFG=D:\CONFIG\DEMO.CFG
```

2

Getting Around in VIEWDAC

This chapter provides an overview of the navigational tools used to get around in the VIEWDAC environment, including menus, dialog boxes, and the VIEWDAC help system. Refer to the *VIEWDAC Reference* manual for more detailed information.

Using VIEWDAC Menus

A menu displays a list of options. VIEWDAC menus are listed horizontally in the menu bar along the top of the screen. To open a menu and display the list of options, either click on the menu's name in the menu bar or press [Alt]+[*x*], where *x* is the highlighted letter in the menu name. For example, Figure 2-1 shows the File menu.

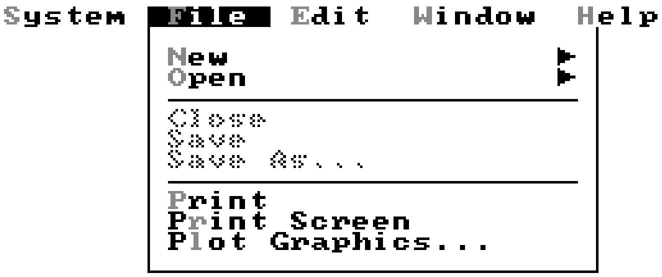


Figure 2-1. File Menu

To select a menu option, either drag the mouse until the desired option is highlighted and then release the mouse button or press the highlighted letter in the option name. When you select an option, the option's name flashes, the menu closes, and the option is executed. To close a menu without selecting an option, click anywhere outside the menu or press [Esc].

Sometimes, menu options are shaded in gray; grayed options are not available for current use. For example, the File\Close, File\Save, and File\Save As options are grayed when no windows are active.

Some menus appear on the menu bar only when certain windows are active. For example, the Graphics menu appears on the menu bar only when the active window is a graphics window.

Refer to Appendix B for a summary of all VIEWDAC menus and the options available in each.

Using VIEWDAC Dialog Boxes

A dialog box is a window used to provide information about a specific operation. A dialog box suspends all system functions until you provide or modify the required information and then close the dialog box. If a dialog box opens while you are performing an operation, you must respond to the dialog box before continuing with the operation. Most dialog box options contain initial responses that you can modify.

You can save the information from several dialog boxes, including the System Configuration dialog box and the Hardcopy Configuration dialog box, in your CFG file. Whenever you change the options in any of these dialog boxes and select the System\Save Options menu option, the CFG file is created (if it does not already exist) or updated (if it does exist). Refer to page 1-13 for more information about CFG files.

Dialog boxes can contain several types of objects; these objects are shown in Figure 2-2.

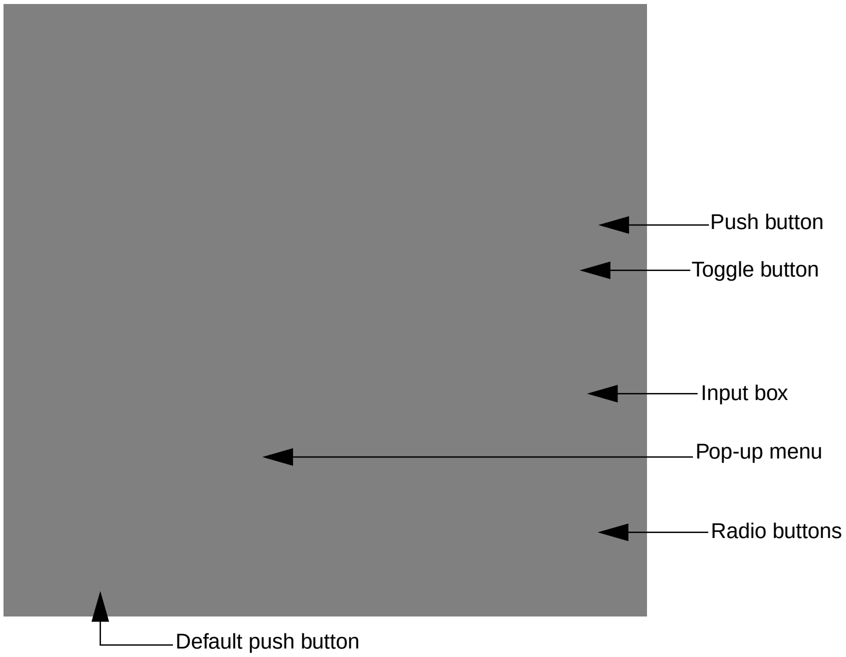


Figure 2-2. Dialog Box Objects

The objects found in dialog boxes are described as follows:

- **Push button -** Use a push button to perform an action. A push button (sometimes called a command button) is represented as an oval button that is labeled with the action it performs. To perform the action, either click on the push button or use [Tab] to highlight the push button and then press [spacebar]. A double thick border denotes the default push button; press [Enter] to select the default push button.

- **Radio button -** Use a radio button to select an option from a set of related options. A radio button (sometimes called an option button) is represented as a small circle next to an option name. Some dialog boxes contain radio icons instead of radio buttons; radio icons use a picture for each option instead of a radio button and an option name.

Only one radio button or icon in a particular set is selected at a time. A black dot in the center denotes the currently selected radio button; a black border denotes the currently selected radio icon. To select a different option, click on the radio button or the option name or click on the radio icon. You can also use [Tab] to highlight the currently selected radio button, and then use the arrow keys to select a different option.

- **Toggle button -** A toggle button enables and disables an option. A toggle button (sometimes called a check box) is represented by a small square box next to the option name. If the box has an X in it, the option is enabled. Click on the box (or on the option name) to alternate the state between enabled and disabled. You can also use [Tab] to highlight the toggle button, and then press [spacebar].

- **Input box -** Use an input box to enter alphanumeric text. An input box (sometimes called a text box) is represented as a rectangular box with a non-shadowed border. Click on the input box or use [Tab] to highlight the input box. Then, enter the appropriate text and either click somewhere outside the input box or press [Tab]. The contents of the input box scroll automatically if the text exceeds the length of the box. String literals are displayed in quotes to distinguish them from data set names.

  Often, a dialog box contains a list box, which displays a list of options or items that you might want to enter in an input box. List boxes provide vertical scroll bars when the number of items exceeds the number of rows displayed in the list box. If a list box is associated with an input box, you can select an item from the list to place that item in the input box.

- **Pop-up menu -** Use a pop-up menu to select an option from a list of related options. A pop-up menu (sometimes called a pop-up list) is represented as a rectangular box with a shadowed border. To select an option from a pop-up menu, click and hold on the box containing the current option to open the pop-up menu (a check mark denotes the current selection), drag the mouse until the desired option is highlighted and then release the mouse button. You can also use [Tab] to highlight the pop-up menu, press [spacebar] to open the pop-up menu, and then either use the arrow keys to highlight the desired option and press [Enter] or press the colored (highlighted) letter (if one is available) in the desired option.

Note: If an object in a dialog box is grayed, it cannot be selected.

Using Shortcut Keys

Table 2-1 lists the shortcut keys you can use to perform certain VIEWDAC operations.

Table 2-1. Shortcut Keys

Key Sequence	Operation
[F1]	Displays context-sensitive help (or help index).
[F2]	Gets context-sensitive Assistance; refer to page 2-10 for more information.
[F3]	Prints the screen to the file or device specified in the Hardcopy Configuration dialog box.
[Ctrl]+[F3]	Prints the screen to a PCX file.
[F4]	Prints the active window or dialog box to the file or device specified in the Hardcopy Configuration dialog box.
[Ctrl]+[F4]	Prints the active window or dialog box to a PCX file.
[F10]	Repaints the screen; exits VIEWRUN emulation mode and turns the menu bar on; closes a modal panel and stops the associated sequence.
[Ctrl]+[X][1]	Cuts (deletes) selected text or object and places text or object on the clipboard.
[Ctrl]+[C][1]	Copies selected text or object and places text or object on the clipboard.
[Ctrl]+[V][1]	Pastes text or object from the clipboard.
[Ctrl]+[D][1]	Duplicates a task (copies and pastes in one step).
[Shift]+Click	Highlights characters between the cursor and the pointer.

Table 2-1. Shortcut Keys (cont.)

Key Sequence	Operation
[Shift]+Double Click	Highlights all words and spaces between the cursor and the pointer.
[Ctrl]+[left arrow]	Moves the text cursor to the beginning of the line.
[Ctrl]+[right arrow]	Moves the cursor to the end of the line.
[Tab]	Moves from one item to the next item in a dialog box.
[Shift]+[Tab]	Moves from one item to the previous item in a dialog box.
[spacebar]	Executes the active item in a dialog box; comparable to clicking on the left mouse button.
[Ctrl]+[spacebar]	Comparable to clicking on the right mouse button.
[Enter]	Executes the default push button in a dialog box.
[Alt]+[*x*]	Selects a menu from the menu bar, where *x* indicates the highlighted letter in the menu name. For example, to select the System menu from the menu bar, press [Alt]+[S]. To select a submenu or a menu option from the displayed menu, press the highlighted letter. For example, to select the Config... option from the System menu, press [Alt]+[S] [C].

Note

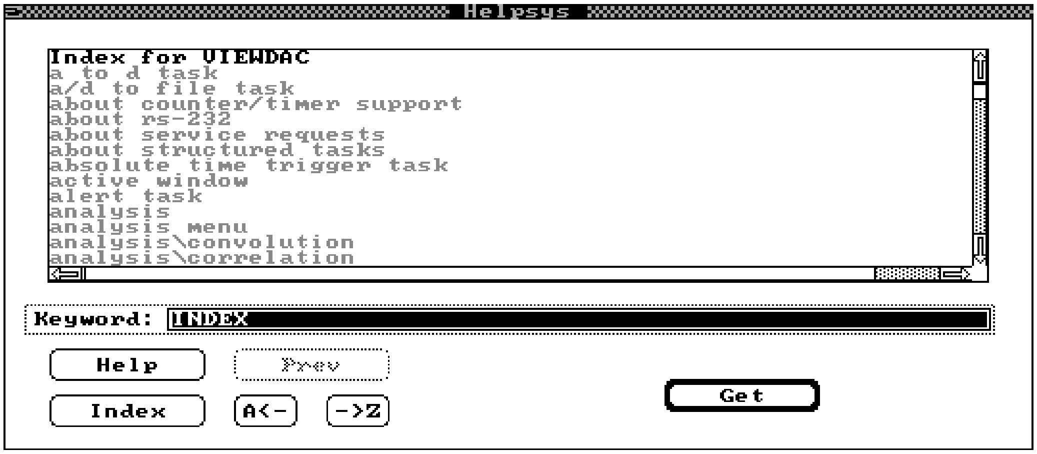
[1] Use these keys when editing input boxes in dialog boxes, since you cannot select the Edit menu when a dialog box is open; use these keys in Loop or Block task dialog boxes to edit the tasks. A bell sounds if you press one of these keys when no text or object is available to cut, copy, paste, or duplicate.

Using the VIEWDAC Help System

The VIEWDAC help system contains an abbreviated version of the material found in the *VIEWDAC Reference* manual and includes information on menus, dialog boxes, windows, commands, and basic concepts.

You can enter the VIEWDAC help system in the following ways:

- **Select Help\Help from the menu bar -** To access the general VIEWDAC help system, select Help\Help from the menu bar. VIEWDAC opens the modeless help window, shown in Figure 2-3.



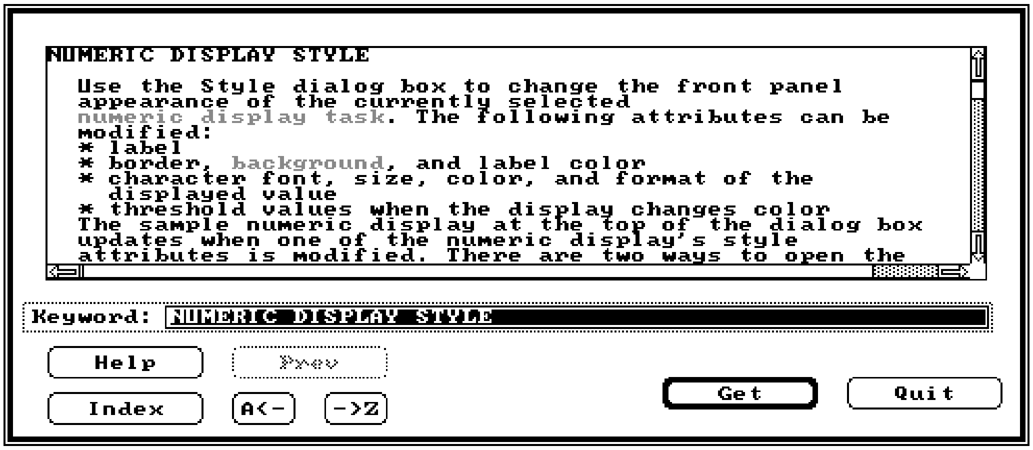
Index for VIEWDAC
a to d task
a/d to file task
about counter/timer support
about rs-232
about service requests
about structured tasks
absolute time trigger task
active window
alert task
analysis
analysis menu
analysis\convolution
analysis\correlation

Keyword: INDEX

Figure 2-3. Modeless Help Window

The modeless help window can remain open while you perform other VIEWDAC functions. You can hide the modeless help window using the Window\Hide menu option. To close the modeless help window, either make the modeless help window active and select the File\Close menu option or click on the close icon in the upper left corner of the window.

- **Press [F1] -** To access information about an active window, menu, or dialog box, press [F1]. VIEWDAC opens a modal help window. For example, if you press [F1] while you are editing the style of a Numeric Display task, VIEWDAC opens a modal help window with information about the Numeric Display Style dialog box; refer to Figure 2-4.



NUMERIC DISPLAY STYLE

Use the Style dialog box to change the front panel
appearance of the currently selected
numeric display task. The following attributes can be
modified:
* label
* border, background, and label color
* character font, size, color, and format of the
  displayed value
* threshold values when the display changes color
The sample numeric display at the top of the dialog box
updates when one of the numeric display's style
attributes is modified. There are two ways to open the

Keyword: NUMERIC DISPLAY STYLE

Figure 2-4. Modal Help Window

A modal help window contains context-sensitive information about the current window, menu, or dialog box. You must close a modal help window before you can perform other VIEWDAC functions. To close a modal help window, select the Quit push button or press [Esc].

Note: If no windows, menus, or dialog boxes are active when you press [F1], VIEWDAC opens the modeless help window.

Both the modeless help window and modal help windows contain a list box that displays the help text, an input box for typing in keywords to look up, and push buttons for navigating through the help system. With the help window open, you can select a particular subject in one of the following ways:

- Type the name of a subject in the Keyword input box and select the Get push button. If no help is available on the subject, VIEWDAC displays information about the next closest alphabetical entry.

- Select the Index push button to display the help index. Scroll through the list of subjects until you find the one you want. Select the subject by either clicking on the entry and then selecting the Get push button or by double-clicking on the entry.

- Click on a red entry in a help window (for color systems) or a highlighted entry in a help window (for monochrome systems) to place the word in the Keyword input box. Then, select the Get push button.

- Double click on a red entry in a help window (for color systems) or a highlighted entry in a help window (for monochrome systems).

- Select the A<- push button to retrieve the previous index entry. Select the ->Z push button to retrieve the next index entry.

- Select the Prev push button to retrieve the previously displayed help information.

To get information on using the help system, select the Help push button.

Getting Assistance

Assistance helps you respond to dialog box input boxes. While using a dialog box input box, either press [F2] or double-click on the input box label to get Assistance. Assistance opens a list box, which contains a list of available options for the operation associated with the dialog box input box. To place one of the available options into the dialog box input box, select the option and then select the OK push button.

For example, suppose you need a pattern for a Numeric Display task front panel. If you are not sure which patterns are available, you can press [F2] to open the Pattern Selection dialog box, which displays a list of available patterns.

Note: Not all dialog box input boxes provide Assistance. If Assistance is not available for the dialog box input box you are using, VIEWDAC responds with a beep when you press [F2].

You can use Assistance along with the VIEWDAC help system. For example, suppose you want to edit the style of a Numeric Display task front panel in the sequence window and you open the Numeric Display Style dialog box. You must supply the number corresponding to a pattern in the Pattern input box but cannot remember which patterns are available.

After selecting the Pattern input box, press [F2] to get Assistance. This opens the Pattern Selection dialog box, which displays a list of available patterns. If you are not sure how to use this dialog box, you can press [F1] to obtain help. The context-sensitive help window opens with the needed information displayed in the list box.

3

VIEWDAC at a Glance

This chapter contains an overview of the features that are specific to VIEWDAC. For more detailed information, refer to the *VIEWDAC Reference* manual.

VIEWDAC Windows

VIEWDAC provides the following types of windows:

- **Sequence window -** Use a sequence window to create, maintain, and run a sequence. A sequence is a collection of tasks and front panels.

  Tasks perform specific functions. For example, an analog input (A to D) task acquires a data point; a Numeric Display task displays a data point. Refer to page 3-4 for more information about VIEWDAC tasks. Front panels present objects to the user; the objects, which you create in your sequence, include push buttons, plots, and numeric displays.

  To open a sequence window, select File\New\Sequence or File\Open\Sequence from the menu bar.

- **Graphics window -** Use a graphics window to graphically plot and analyze numeric data sets. You can zoom in on any section of the plotted data to create another graphics window called the zoom window. You can save both the graphical display and the data to a data file, if desired.

  To open a graphics window, select File\New\Graphics or File\Open\Graphics from the menu bar.

- **Table editor window -** Use a table editor window to view the elements of a data set in a tabular format, alter the values in a data set, and perform calculations on the values in a data set. You can import data from another application program and export data from a table editor window into another application program.

  To open a table editor window, select File\New\Table Edit or File\Open\Table Edit from the menu bar.

- **Text editor window -** Use a text editor window to create and edit ASCII text documents.

  To open a text editor window, select File\New\Text Edit or File\Open\Text Edit from the menu bar.

- **Waveform generator window -** Use the waveform generator window to create periodic waveforms, such as square, triangular, and sawtooth waves, and uniform and normal noise waveforms. You can manipulate waveforms and combine several waveforms to create a composite waveform.

  To display the waveform generator window, select Window\Wave Gen from the menu bar.

- **Clipboard view window -** Use the clipboard view window to view the data on the clipboard.

  To display the clipboard view window, select Window\Clipboard from the menu bar.

- **Help window -** Use a help window to provide information about VIEWDAC features. Refer to page 2-7 for more information.

- **RS-232 terminal window -** Use the RS-232 terminal window to communicate with an RS-232 device without using sequence tasks.

  To display the RS-232 terminal window, select Window\Terminal from the menu bar.

When multiple windows are open simultaneously, the title bar of the active window has a checkerboard background, while the title bars of the other windows have a solid background. To make a window active, click on the window or select Window\Window List... from the menu bar and select the appropriate window.

The menu bar reflects the active window; only menus associated with the active window's type are available. For example, the Graphics menu is available only when the active window is a graphics window.

You can interactively resize graphics windows, table editor windows, text editor windows, the clipboard view window, and front panels by dragging their borders with the pointer. (Note that you can resize front panels and front panel objects only when the associated sequence is not running.) You cannot interactively resize sequence windows, the waveform generator window, or help windows.

To modify window attributes, such as size, position on the screen, and background color, select Window\Preferences from the menu bar and make the appropriate changes. You can then save these changes by selecting System\Save Options.

To close the active window, click on the close icon in the upper left corner of the window or select File\Close from the menu bar. To avoid losing the contents of a window, you should save the window before closing it. (Note that closing the clipboard view window does not delete its contents.)

Some VIEWDAC windows, such as sequence windows and graphics windows, are associated with unique file names; more than one of this type of window can be open simultaneously. Some VIEWDAC windows, such as the clipboard view window and the modeless help window, are system utilities; only one of each type of system utility window exists in VIEWDAC.

Normally, VIEWDAC windows are independent of one another. However, sometimes windows exist in a parent/child relationship. A child window is a window that exists only when another window (the parent window) is open. Examples of child windows are zoom windows, which display an expanded portion of a parent graphics window, and front panels, which are always associated with a parent sequence window. When you close a parent window, its child window automatically closes.

VIEWDAC Tasks

This section describes VIEWDAC tasks, arranged according to their functions.

System Tasks

System tasks are general maintenance tasks that you can use with any VIEWDAC sequence. Note that certain System tasks are referred to as structured tasks. A structured task contains a series of other tasks. Tasks within a structured task execute in a sequential order.

The System tasks are described as follows:

- **Loop task -** A structured task that allows two or more tasks to run consecutively. A loop counter determines how many times the task list is executed. A Loop task can run its task list infinitely, run it a specified number of times, or run it at a certain rate. You can simplify most sequences by using one or more loops; you can use loops inside of other loops.

- **Block task -** A structured task that executes its task list once. A Block task is more convenient to use than a Loop task since you do not have to set loop timing or iteration values.

- **Begin Until task -** A structured task that executes its task list once and then checks a condition. If the condition is not true, the tasks are repeated; if the condition is true, the Begin Until task stops. A Begin Until task always executes at least once. If the possibility exists that the condition will never be true (data value remains zero), you must use another task (such as a Push Button task) to stop the Begin Until task.

- **Do While task -** A structured task that checks a condition first. If the condition is not true, the Do While task stops. If the condition is true, the Do While task executes its task list sequentially and the condition is checked again. A Do While task may never execute. If the possibility exists that the condition will always be true (data value remains non-zero), you must use another task (such as a Push Button task) to stop the Do While task.

- **Call task -** Starts a top-level Block task. A Call task is useful if you want to use the same set of instructions more than once in your VIEWDAC sequence.

- **NOP task -** Starts or stops other tasks. A NOP (No OPeration) task runs its control list when it is started and when it is stopped. NOP tasks are run in the next available foreground time. When a NOP task runs, it returns control to the foreground; this ensures that front panels (and other tasks that use the foreground) have time to update when pre-emptive mode is enabled.

- **Macro task -** Executes a VIEWDAC macro. A Macro task can link the sequence to a non-sequence VIEWDAC window, such as a graphics window. Macro tasks are useful for performing often-repeated keystrokes.

- **Tone task -** Provides an audio alarm. For example, a Tone task started by another task sounds when the first task finishes running.

- **Alert task -** Conditionally displays a customized message in a dialog box.

- **Fix DOS Time task -** Resets the VIEWDAC clock and the DOS clock to the computer's CMOS clock time and date. This allows you to keep the VIEWDAC and DOS clocks up-to-date during long block-mode acquisition tasks.

- **Lock task -** Locks the VIEWDAC environment so that VIEWDAC no longer responds to the mouse or to keyboard input. You unlock the VIEWDAC environment by running the Unlock task in the sequence.

- **Unlock task -** Unlocks the VIEWDAC environment. The Unlock task must be started by a task that does not require user input, such as a Time Trigger task; the Unlock task cannot be started by a task that requires user input, such as a Push Button task.

Panel Tasks

Panel tasks place an object in a front panel either for display or to interact with the sequence as it is running. The Panel tasks are described as follows:

- **Numeric Display task -** Places a numeric display on the front panel to monitor the value of a scalar numeric data set.

- **Numeric Input task -** Places an input box on the front panel. The input box allows you to interactively modify the value of a scalar numeric data set while a sequence is running.

- **String Display task -** Places a string display on the front panel to display a scalar string data set.

- **String Input task -** Places an input box on the front panel. The input box allows you to interactively modify the value of a single scalar string data set while a sequence is running.

- **Array Display task -** Places an array display on the front panel to monitor the value of an array data set.

- **Meter task -** Graphically displays the current value of a scalar data set on the front panel. The Meter task is useful for displaying data typically displayed by gauges, such as temperature, pressure, and voltage.

- **Strip Chart task -** Graphically displays the variation of one or more scalar numeric data sets relative to another scalar numeric data set, such as time.

- **Line Graph task -** Graphically displays one or more numeric array data sets.

- **Binary Input task -** Places a toggle button on the front panel. The toggle button allows you to interactively modify the value of a scalar data set.

- **Bar Graph task -** Graphically displays the value of one or more scalar numeric data sets using rectangles or bars. The length of the bar is proportional to the value of the data set it represents.

- **Hide/Show task -** Either shows or hides a front panel while the sequence is running. Hide/Show tasks are useful for modal front panels, which do not contain a close icon or a stop push button.

- **Region Fill task -** Sets a rectangular area on a front panel to a specified color. Region Fill tasks are useful for matching the background color of a front panel to the color of a PCX file used in the background of a front panel or for clearing a front panel at a specified time.

Trigger Tasks

Trigger tasks start and stop other tasks either interactively or conditionally. The Trigger tasks are described as follows:

- **Push Button task -** Places a push button on the front panel. The Push Button task starts or stops other tasks when the push button is pressed.

- **Toggle Button task -** Places a toggle button on the front panel. The toggle button acts as an ON/OFF switch. If the toggle button is in the ON position, the Toggle Button task runs its control start list; if the toggle button is in the OFF position, the Toggle Button task runs its control stop list.

- **Time Trigger task -** Waits a specified amount of time before it runs its control list. Use a Time Trigger task to add a time delay to sequence execution.

- **Absolute Time Trigger task -** Waits until a specified time and/or date before it runs its control list. Use an Absolute Time Trigger task to start and stop other tasks at a specific time and/or date.

- **Comparison task -** Compares a data point to one or two specified values. Comparison tasks are useful for If/Then and If/Then/Else statements.

- **Threshold task -** Determines when a changing data set crosses a specified point and whether the data set is increasing or decreasing in value. The Threshold task is useful for determining rising and falling edges.

- **Test Bits task -** Performs a bitwise AND of two data sets. If the result is 1, the Test Bits task runs its control list; if the result is 0, the control list is not run. The Test Bits task is useful for conditionally controlling other tasks.

Calculation Tasks

Calculation (Calc) tasks perform calculations on data. The Calc tasks are described as follows:

- **Expression task -** Creates an expression that uses one or more function. Refer to Appendix C for a summary of the operators and mathematical functions that you can use in these expressions. Using an Expression task is simpler than stringing several other Calc tasks together.

- **PID Control task -** Monitors and adjusts a process while the process is running. You can use a Numeric Input task to change the values of the setpoint and the PID (proportional integral derivative) gains while the PID Control task is running.

- **FFT task -** Performs an FFT (Fast Fourier Transform) on a data set.

- **IFFT task -** Performs an IFFT (Inverse FFT) on a data set. Typically, the IFFT task is used with the FFT task.

- **Numeric Operation task -** Performs operations typically found on a calculator; you can perform simple arithmetic operations, simple unary functions, and random number generation on either a scalar data set or an array.

- **Array Operation task -** Performs array management and calculation operations on an array.

- **Trig Operation task -** Performs trigonometric and hyperbolic operations (and the inverses of these functions) on either a scalar data set or an array.

- **Statistic Operation task -** Performs statistical operations on a data set of two or more elements.

- **Bitwise Operation task -** Performs Boolean operations on binary data.

- **Logical Operation task -** Compares either two numbers, a number and an array, or two arrays. The result of the comparison is 0 if the comparison is false or 1 if the comparison is true; you can use this result with Bitwise Operation tasks to create complex logical expressions using AND and OR operations.

- **String Operation task -** Manipulates and performs calculation operations on string data sets.

- **Polynomial Evaluation task -** Converts either a scalar data set or an array from one form to another by applying a polynomial equation to the data.

- **Waveform Average task -** Creates an average waveform from several data sets. One wave at a time is averaged to allow inspection of convergence. The Waveform Average task is useful in increasing the signal-to-noise ratio in noisy waveforms.

- **Moving Average task -** Calculates the average of a specified number of data points. The Moving Average task is useful for filtering data quickly.

- **Moving Median task -** Calculates the moving median of a specified number of data points. The Moving Median task is useful for removing noise spikes from data.

- **Moving Filter task -** Filters a specified number of data points. The Moving Filter task is useful for filtering data thoroughly.

- **Moving Derivative task -** Calculates the moving derivative of a specified number of data points.

- **Running Integral task -** Calculates the running integral of input data points.

- **Moving Peak Find task -** Finds peaks (high points) in a specified number of data points. The Moving Peak Find task is useful for triggering other tasks when a peak is found.

- **Moving Level Find task -** Finds places where a data set crosses a specified value. The Moving Level Find task is useful for triggering other tasks when a crossing is found.

- **Moving Valley Find task -** Finds valleys (low points) in a specified number of data points. The Moving Peak Find task is useful for triggering other tasks when a valley is found.

- **Array Filter task -** Filters an input data set with a low-pass, high-pass, band-pass, or band-stop filter.

- **Array Peak Find task -** Finds peaks (high points) in a data set by applying a quadratic equation to successive sets of data points. A peak is found when the applied quadratic is concave downward.

- **Array Level Find task -** Finds places where a data set crosses a specified value by applying a linear equation to successive sets of data points.

- **Array Valley Find task -** Finds valleys (low points) in a data set by applying a quadratic equation to successive sets of data points. A valley is found when the applied quadratic is concave upward.

- **Array Freq Estimation task -** Performs frequency estimation calculations using a Fourier analysis of a data set.

- **Pulse Rise Parms task -** Calculates the rise time of a rising edge.

- **Pulse Fall Parms task -** Calculates the fall time of a falling edge.

- **Pulse Parms task -** Calculates the amplitude of the top and the bottom of a pulse.

Data I/O Tasks

Data I/O tasks write data to and read data from data files on disk, VIEWDAC arrays, computer I/O ports, and computer memory (RAM). The Data I/O tasks are described as follows:

- **DOS task -** Performs DOS file operations, such as copy, delete, and rename, on a specified file. A DOS task allows you to manipulate files during sequence execution.
- **Array Write task -** Writes numeric scalar data to an array.
- **Array Read task -** Reads numeric scalar data from an array.
- **ASCII Write task -** Writes alphanumeric scalar and array data to an ASCII file.
- **ASCII Read task -** Reads numeric scalar and array data from an ASCII file.
- **Binary Write task -** Writes alphanumeric scalar and array data to a binary file.
- **Binary Read task -** Reads alphanumeric scalar and array data from a binary file.
- **VIEWDAC Write task -** Writes alphanumeric scalar and array data to a VIEWDAC file.
- **VIEWDAC Read task -** Reads alphanumeric scalar and array data from a VIEWDAC file.
- **ASYST Write task -** Writes numeric array data to an ASYST file.
- **ASYST Read task -** Reads numeric array data from an ASYST file.
- **ASYSTANT Write task -** Writes numeric array data to an ASYSTANT file.
- **ASYSTANT Read task -** Reads numeric array data from an ASYSTANT file.

- **Port I/O task -** Writes data to or reads data from a specified port. The Port I/O task is useful when using non-standard or custom-built boards in your computer and when accessing I/O-mapped devices.

- **Poke/Peek task -** Writes to or reads from the real-mode memory (lower 1M byte) in your computer. The Poke/Peek task is useful when accessing memory-mapped devices.

DAS Tasks

DAS tasks perform analog I/O operations, digital I/O operations, and counter/timer I/O operations. The DAS tasks are described as follows:

- **A to D task -** Sets up an analog input operation.

- **A/D to File task -** Sets up an analog input operation and stores the data directly in a binary data file.

- **Thermocouple task -** Sets up an analog input operation that acquires temperature data from a thermocouple device.

- **D to A task -** Sets up an analog output operation.

- **Digital Input task -** Sets up a digital input operation.

- **Digital Output task -** Sets up a digital output operation.

- **Frequency Measurement task -** Counts the number of pulses (events) during a specified gate period.

- **Event Counting task -** Continually counts pulses (events) until the Event Counting task is stopped.

- **Pulse Output task -** Sets up a pulse stream or single pulse with a specified pulse width and duty cycle.

- **Time Interval task -** Measures a time duration (the time between the start and the end of a process).

GPIB Tasks

GPIB (General Purpose Interface Bus) tasks write data to and read data from GPIB devices. The GPIB tasks are described as follows:

- **GPIB Read task -** Reads ASCII or binary data from a GPIB device into one or more data sets.
- **GPIB Write task -** Writes ASCII or binary data to one or more GPIB devices.
- **Serial Poll task -** Consecutively polls all specified GPIB devices on the specified GPIB.
- **SRQ Test task -** Determines whether any GPIB device on the specified GPIB has asserted the SQR (service request) line.
- **SRQ Wait task -** Waits for a SRQ to occur on any GPIB device on the specified GPIB.
- **Reset Bus task -** Resets the specified GPIB.
- **Send Interface Clear task -** Stops all activity on the specified GPIB.
- **REN On task -** Asserts the REN (remote enable) line on the specified GPIB.
- **REN Off task -** Unasserts the REN line on the specified GPIB.
- **Device Clear task -** Forces all GPIB devices on the specified GPIB that are capable of responding into a predefined, device-dependent clear state.
- **Selected Device Clear task -** Forces selected GPIB devices on the specified GPIB that are capable of responding into a predefined, device-dependent clear state.
- **Go to Local task -** Forces selected GPIB devices on the specified GPIB into local panel control.
- **Local Lockout task -** Forces all GPIB devices that are capable of responding to ignore all front panel (local) commands.

- **Go to Standby task -** Unasserts the ATN (attention) line and puts VIEWDAC into the controller standby state.

- **Group Execute Trigger task -** Forces selected GPIB devices on the specified GPIB to initiate a device-dependent event, such as data logging.

RS-232 Tasks

RS-232 tasks write data to and read data from serial devices using the RS-232 communications protocol. The RS-232 tasks are described as follows:

- **RS-232 Read task -** Reads ASCII or binary data from a specified RS-232 port into one or more data sets.

- **RS-232 Write task -** Writes ASCII or binary data to one or more serial devices using the RS-232 communications protocol.

- **RS-232 DTR/RTS task -** Asserts the DTR (data transmit ready) line and/or the RTS (request to send) line.

- **RS-232 CTS/DSR task -** Monitors the CTS (clear to send) line, the DSR (data set ready) line, the CD (carrier detect) line, and/or the RI (ring indicator) line.

- **RS-232 Flush task -** Clears the VIEWDAC transmit (Tx) buffer and/or the VIEWDAC receive (Rx) buffer.

Printing in VIEWDAC

This section provides information about printing in VIEWDAC.

Types of Printing

VIEWDAC provides the following types of print images:

- **Bitmap print -** A bitmap print is an exact duplicate of the image on the screen, pixel by pixel. A bitmap print is also called a screen dump, screen print, screen capture, bitmap dump, or print dump.

  Bitmap prints consist of dots, so the images tend to be more jagged. The size of the printed image depends on the resolution of your printer. You can scale the image to a larger or smaller size, if desired; however, if you do, the printing process is slower.

  You can produce a bitmap print of a text window, a graphics window, or an entire screen, using a printer hardcopy device. Supported printers are listed in the Hardcopy Configuration dialog box.

  Press [F4] to produce a bitmap print of any VIEWDAC window (the entire window is printed). Select File\Print from the menu bar to produce a bitmap print of a front panel (the entire window is printed) or to produce a bitmap print of the contents only of a graphics window.

  Note: For sequence, text editor, and table editor windows, selecting File\Print from the menu bar prints the text contents of the window, not a bitmap print.

  To produce a bitmap print of the entire screen, select File\Print Screen from the menu bar or press [F3].

- **Plotter output -** Plotter output is a representation of the image on the screen using lines and characters; it is not an exact match. Plotter output is also called vector output, line output, or line graphics.

Plotter output consists of lines, so the images tend to be smoother and the printing process is faster. You can scale the image, if desired, without slowing down the printing process.

You can produce plotter output of a graphics window only, using a plotter hardcopy device. Supported plotters are listed in the Hardcopy Configuration dialog box.

To plot a graphics window, select File\Plot Graphics from the menu bar. Note that the background of the graphics window is not printed.

For both bitmap prints and plotter output, VIEWDAC uses the information in the Hardcopy Configuration dialog box to control the appearance of the output. Refer to the *VIEWDAC Reference* manual and *VIEWDAC - What's New in Version 2.2* for more information about printing.

If your printer supports PostScript mode or Hewlett-Packard Graphics Language (HPGL) mode, you can use your printer to produce plotter output. Select the Plotter radio button in the Hardcopy Configuration dialog box. For PostScript mode, select PostScript or Color PostScript from the list box. For HPGL mode, select one of the HP plotters listed. Any of the listed HP plotters should work; however, it is recommended that you select the one that is closest to your printer in such things as page size and number of pens.

Note: If your printer does not automatically switch between bitmap output and plotter output, you may have to manually adjust your printer hardware or software. Refer to your printer documentation for more information.

Printing to Files

You can send printer output to a DOS file. For example, you can print a sequence window to a DOS file to produce a hardcopy description of your sequence. You can also use plotter output to send encapsulated PostScript (EPS) to an EPS file by selecting PostScript as the plotter hardcopy device. This allows you to use desktop publishing programs to incorporate plots into documents.

4

Troubleshooting

Table 4-1 lists some common problems you may encounter when using VIEWDAC and possible solutions to these problems. If you encounter a problem that you cannot solve on your own, contact the Keithley MetraByte Applications Engineering Department.

Table 4-1. Common Problems

Problem	Possible Cause	Possible Solution
Erroneous data or unexpected results	You are attempting to acquire data at a rate faster than your board can handle.	Check the specifications of your board. If you are performing an RS-232 or interrupt-driven operation, try using the keyboard instead of the mouse.
	You are using an invalid data type in a binary read operation, such as the Table\Import Data\Binary operation or the Binary Read task.	Make sure that data types match. For example, read floating point data into floating point variables; read integer data into integer variables.
	You exited to DOS using System\DOS Shell while a foreground task was still running.	Make sure that all foreground tasks have finished running before you exit to DOS.
	You are using a keyboard accelerator program, such as Repeat Performance by WordPerfect, that is not compatible with VIEWDAC.	Remove all keyboard accelerator programs to see if your problems are solved.
	A system interrupt, such as the mouse or disk I/O, is interfering with a fast, interrupt-driven data acquisition.	Try using the keyboard instead of the mouse.

Table 4-1. Common Problems (cont.)

Problem	Possible Cause	Possible Solution
Erroneous data or unexpected results (cont.)	You are using macros from a previous version of VIEWDAC.	Rerecord your macros.
System failure	You are performing an A to D, D to A, Digital Input, or Digital Output task in Block Transfer Mode with either External Clocking or External Triggering enabled, but your board cannot recognize the external clock or external trigger.	Check the hardware to make sure that the external clock or trigger is connected correctly. Make sure that the external clock provides the correct number of ticks. Use either DMA Transfer Mode or Interrupt Transfer Mode.
	You are performing a GPIB Read or GPIB Write task with DMA enabled and either timeouts are not occurring or the GPIB device is not responding.	Debug your GPIB sequences without using DMA; enable DMA when the sequences are running properly.
	You are using an invalid data type in a binary read operation, such as the Table\Import Data\Binary operation or the Binary Read task.	Make sure that data types match. For example, read floating point data into floating point variables; read integer data into integer variables.
System failure during repeated hard disk drive operations	You are using the EMM386 expanded memory manager without the DOS utility SMARTDrive.	Install SMARTDrive. Refer to your DOS documentation for more information.
Problems starting VIEWDAC	Your computer does not have an appropriate math coprocessor.	Make sure that your computer has a math coprocessor and that it is 100% Intel-compatible. Note that the 486SX (and some other 486 microprocessors) do not have a built-in math coprocessor.
Problems with mouse	On some faster computers, the mouse may not respond correctly to rapid movements in VIEWDAC.	Refer to your mouse documentation for information on adjusting the sensitivity of your mouse. If you have an available serial port, use the mouse driver provided with VIEWDAC.

Table 4-1. Common Problems (cont.)

Problem	Possible Cause	Possible Solution
Problems with mouse (cont.)	An application program running on a computer with a 16550 UART chip has enabled the UART chip's FIFOs. (Typically, the 16550 UART chip is found on newer-model computers.)	Use Microsoft Diagnostics to determine which UART chip is used in your computer. (Enter `MSD` at the DOS prompt and select Com Ports from the main menu.) If your computer uses a 16550 UART chip, disable the FIFOs by including the line **disable.uart.fifos** in your initialization file. Refer to Appendix A of the *VIEWDAC Reference* manual for more information about initialization files.
Problems with screen	Some DOS screen savers may interfere with VIEWDAC operations.	Disable the screen saver you are using. Try a different screen saver.
Problems with printing or plotting	You pressed [Ctrl]+[C] while plotting to an HP7475.	Wait until the plot to the HP7475 has finished and try again.
	You are plotting to a WordPerfect file (using File\Plot Graphics, with Output set to Postscript and EPS enabled), and the file cannot read the plot.	Open the EPS file with a text editor, locate the line **%!PS-Adobe-1.0** at the top of the file, and manually change **1.0** to **2.0**.
Out-of-memory errors	VIEWDAC requires additional memory.	Save your work, exit from VIEWDAC (using System\Bye), and restart VIEWDAC. If you continue to encounter out-of-memory errors, make sure that you have sufficient RAM in your system (minimum of 6M bytes). Make sure that you are running VIEWDAC in virtual memory mode.
Overrun errors	You are attempting to acquire data at a rate faster than your board can handle.	If you are performing an RS-232 or interrupt-driven operation, try using the keyboard instead of the mouse.

Table 4-1. Common Problems (cont.)

Problem	Possible Cause	Possible Solution
Frequent parity errors	Parity error checking is interfering with a fast DMA acquisition.	Disable parity error checking by including one of the following lines in your initialization file: **0C0%h 0C%h set.nmi.mask** (this should work for most computers) **FF%h 0%h set.nmi.mask** (this is required for Zenith machines) Refer to Appendix A of the *VIEWDAC Reference* manual for more information about initialization files.
VIEWDAC cannot find your RS-232 port	An application program running on a computer with a 16550 UART chip has enabled the UART chips's FIFOs. (Typically, the 16550 UART chip is found on newer-model computers.)	Use Microsoft Diagnostics to determine which UART chip is used in your computer. (Enter `MSD` at the DOS prompt and select Com Ports from the main menu.) If your computer uses a 16550 UART chip, disable the FIFOs by including the line **disable.uart.fifos** in your initialization file. Refer to Appendix A of the *VIEWDAC Reference* manual for more information about initialization files.

A

Memory Usage in VIEWDAC

This appendix contains some general information about the way VIEWDAC uses the memory in your computer. For more information, refer to *Extending DOS* by Duncan, et al (Addison-Wesley Publishing Co.).

Random Access Memory

To run VIEWDAC, your system typically consists of the following configuration of RAM:

- **640K bytes of conventional memory -** This is used for DOS, for PharLap's 386/DOSExtender (which allows VIEWDAC to communicate with DOS), for the Keithley external DAS drivers (which allow VIEWDAC to communicate with the Keithley DAS boards in your computer), for the memory buffers (which are required for operations performed in interrupt transfer mode or DMA transfer mode), and for any tools or utilities you want to run.

- **384K bytes of high memory -** This is typically used for video and network adapters and for the BIOS that helps DOS run your computer.

- **A minimum of 5M bytes of extended memory -** This is used for everything that is not stored in conventional memory or high memory. Most computers can access up to 16M bytes of extended memory; many computers, including Micro Channel™-bus and EISA-bus computers, can access more than 16M bytes of extended memory. VIEWDAC can use all the extended memory in your computer; once VIEWDAC allocates extended memory, it is not available for other purposes until you exit VIEWDAC.

Expanded Memory Managers

If desired, you can use an expanded memory manager (EMM), such as QEMM/386, 386MAX, or EMM386, to load terminate-and-stay-resident (TSR) programs, such as the Keithley external DAS drivers, into high memory; this ensures that you will have enough conventional memory for your other needs. Refer to your EMM documentation to learn how your EMM works.

Considerations

If you are using an EMM with VIEWDAC, keep the following in mind:

- Make sure that your EMM provides VCPI support. VCPI is used by the EMM and your computer to share memory.

- To allow the EMM to recognize memory-mapped devices, such as the Keithley 500 Series, make sure that you exclude the memory regions where the memory-mapped devices reside.

- Allow the EMM to select the location of the expanded memory frame; this will maximize the amount of high memory available.

- Configure the EMM to provide all memory as expanded memory, rather than extended memory; when VIEWDAC needs memory, the EMM converts expanded memory to extended memory automatically.

- If you plan to perform operations in DMA transfer mode, do not allow the EMM to rearrange memory. For example, if you are using QEMM/386, make sure that you use the NOSORT option.

- On some computers, it is more efficient if you use your computer’s setup program to provide SHADOW BIOS functions instead of using the EMM’s SHADOW BIOS functions. Try both methods to see which provides the most extended memory.

Setup Examples

This section contains two setup examples. Note that these examples may not be appropriate for your particular computer. Refer to your DOS documentation and to your EMM documentation for more information.

The following are examples of setup files for QEMM/386 that takes advantage of its ability to load device drivers and DOS buffers into high memory.

CONFIG.SYS File

```
DEVICE=C:\QEMM\QEMM386.SYS ON HANDLES=255 RAM NOSORT X=CFF0-CFFF
buffers = 1
files = 10
stacks = 0,0
break = on
shell=command.com /e:300 /p
```

AUTOEXEC.BAT File

```
@echo off
c:\qemm\loadhi c:\qemm\buffers = 40
c:\qemm\loadhi c:\qemm\files = 25
c:\qemm\loadhi d:\util\mouse.com

path c:\dos;c:\VIEWDAC;d:\util
```

The following are examples of setup files for 386MAX that takes advantage of its ability to load device drivers and DOS buffers into high memory. 386MAX can use a profile file; in this example, the profile file is 386MAX.PRO. Note that EMS must be ON in order for 386MAX to provide VCPI services.

CONFIG.SYS File

```
Device=c:\386max\386max.sys pro=c:\386max\386max.pro
files=20
buffers=1
stacks=0,0
```

386MAX.PRO File

```
                  ; VGA
RAM=C600-CB00     ; use for VIDEO7 VRAM VGA
USE=B000-B800     ; INSTALL ==> recover RAM in MDA region
USE=F000-F800     ; INSTALL ==> recover fill regions in ROM
USE=FD00-FE00     ; INSTALL ==> recover fill regions in ROM
                  ; This profile created automatically by INSTALL
PRGREG=3          ; Load 386MAX.SYS into this program region
```

AUTOEXEC.BAT File

```
prompt $p$g
path c:\dos;c:\386max;c:\util
c:\386max\386load size=65792 prgreg=2 flexframe prog=c:\dasdemo
c:\386max\386load size=28032 prgreg=2 flexframe prog=ipx-nov
c:\386max\386load size=41984 prgreg=2 flexframe prog=net3
c:\386max\386load size=14768 prog=c:\util\mouse
```

B

Summary of VIEWDAC Menus

This appendix provides a summary of all the VIEWDAC menus available from the menu bar and the options available with each. Table B-1 lists the VIEWDAC menus alphabetically and indicates the figure number and page number in this appendix where each menu is shown.

Table B-1. VIEWDAC Menus

Menu	Figure and Page Number
Analysis	Figure B-11, page B-12
Edit	Figure B-3, page B-5
File	Figure B-2, page B-4
Graphics	Figure B-8, page B-9
Help	Figure B-5, page B-7
Panel	Figure B-7, page B-8
Parameter	Figure B-12, page B-13
Search	Figure B-9, page B-10
Sequence	Figure B-6, page B-7
System	Figure B-1, page B-3
Table	Figure B-10, page B-11
Terminal	Figure B-13, page B-14
Window	Figure B-4, page B-6

Keep the following in mind as you use VIEWDAC menus:

- Some menus appear on the menu bar only when certain types of windows are active; these menus are noted in this appendix.

- Some menu options are not available for use at particular times. If a menu option is not available, the menu option is grayed on the screen.

- Some menu options are followed by an ellipsis (. . .); selecting one of these options opens a dialog box that requires a response.

- Some menu options are followed by an arrowhead (➤); selecting one of these options opens a submenu. In this appendix, these menu options are followed by an arrow pointing to the appropriate submenu.

- In each menu option, one of the letters (or numbers) is underlined. To select a menu from the menu bar, you can hold down [Alt] while you press the underlined letter (or number); to select a menu option from one of the menus, you can press just the underlined letter (or number). For example, to select the File\Open\Sequence menu option, you can press [Alt]+[F] [O] [S].

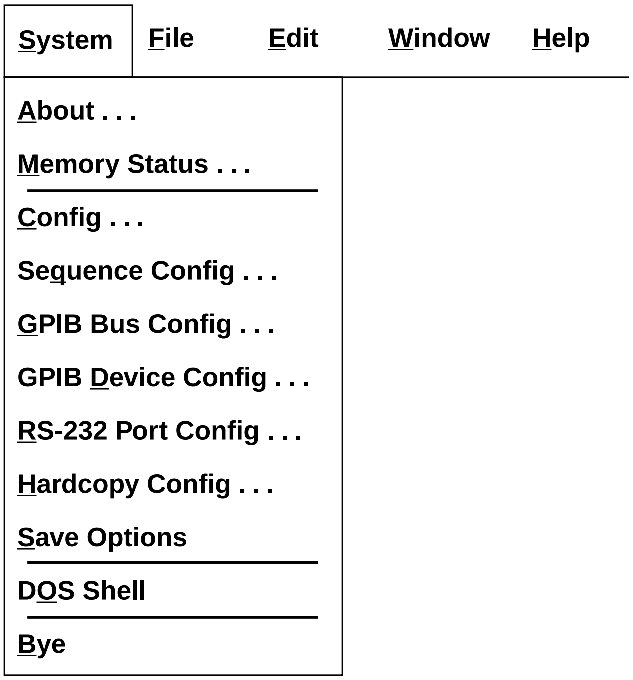


Figure B-1. System Menu

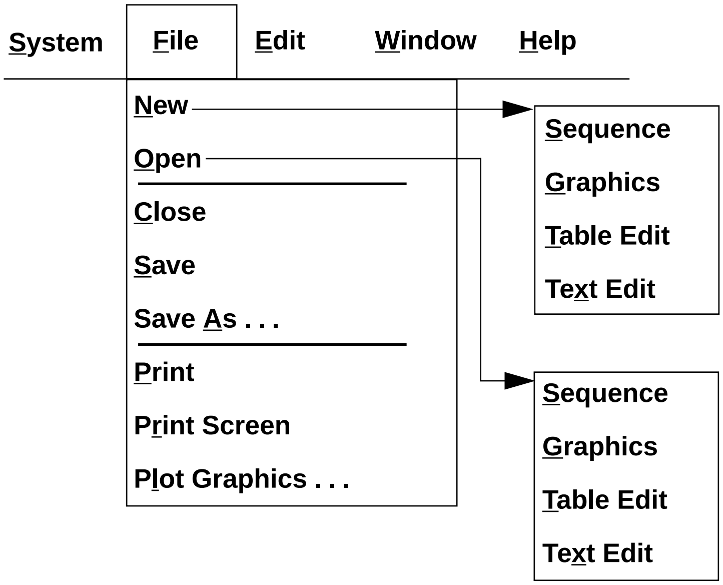
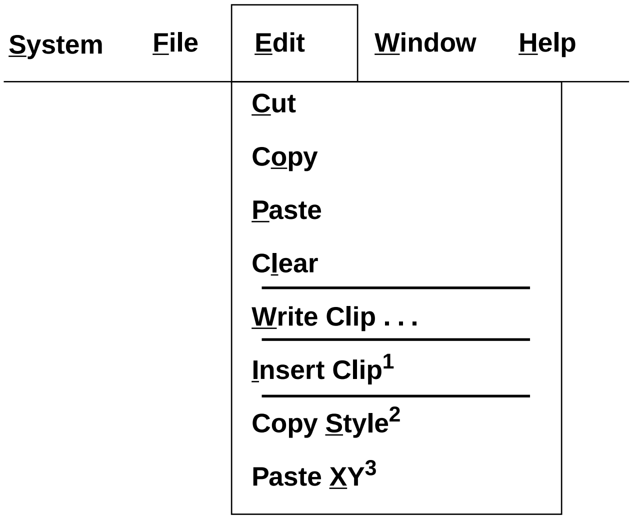


Figure B-2. File Menu



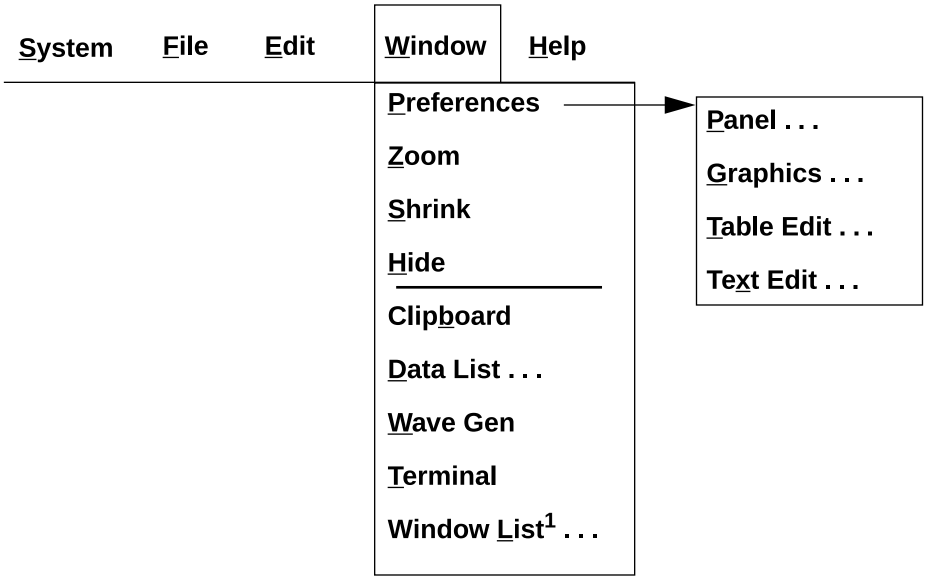
Notes:

[1] Only available when a table editor window is active and the clipboard contains data.

[2] Only available when a graphics window is active.

[3] Only available when a graphics window is active and two or more data sets of the same size have been copied from a table editor window or a graphics window to the clipboard.

Figure B-3. Edit Menu



Note:
[1] List of all windows follows. A check mark indicates the active window.

Figure B-4. Window Menu

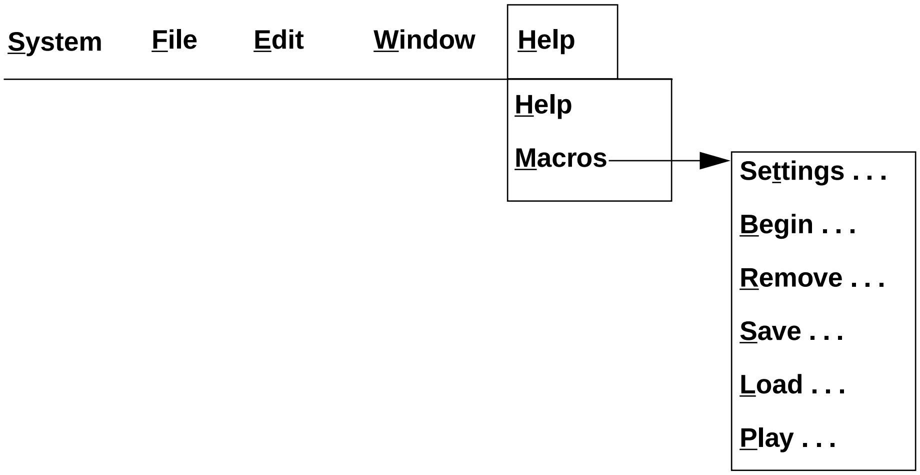
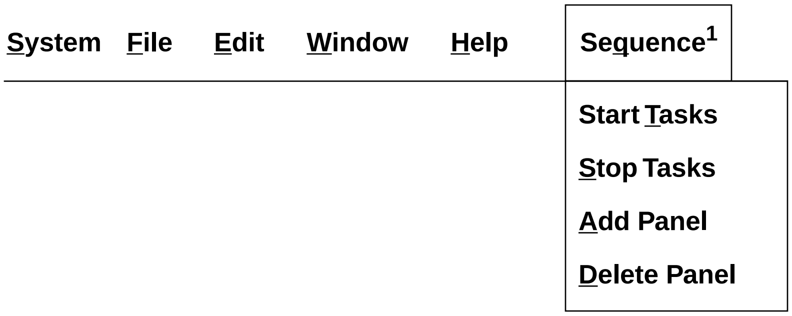
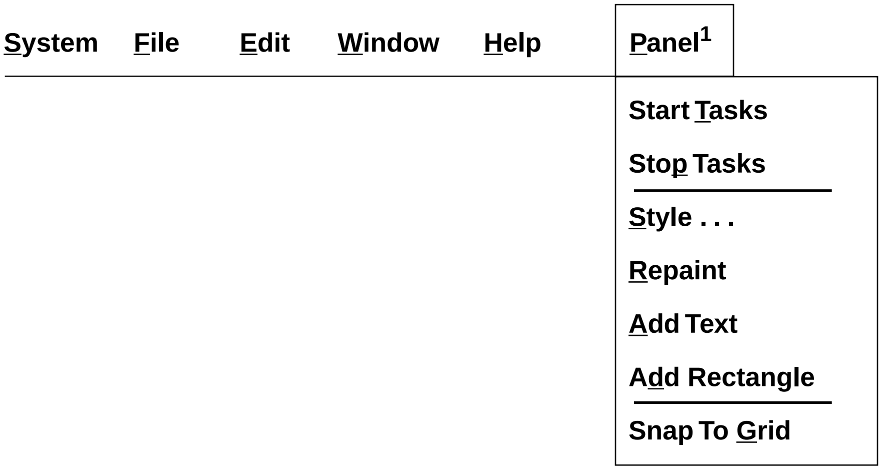


Figure B-5. Help Menu



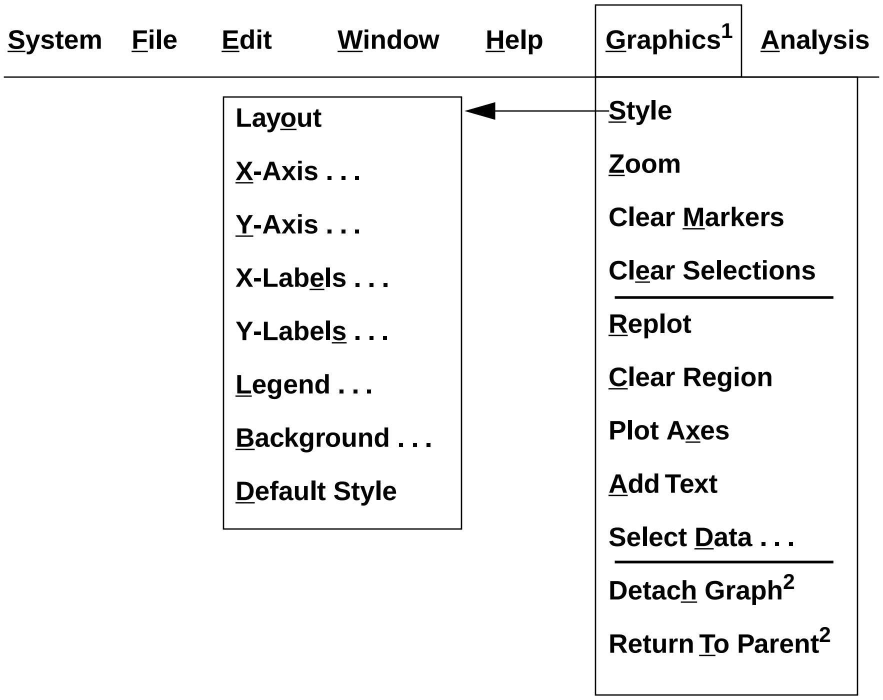
Note:
[1] Only available when a sequence window is active.

Figure B-6. Sequence Menu



Note:
[1] **Only available when a front panel window is active.**

Figure B-7. Panel Menu

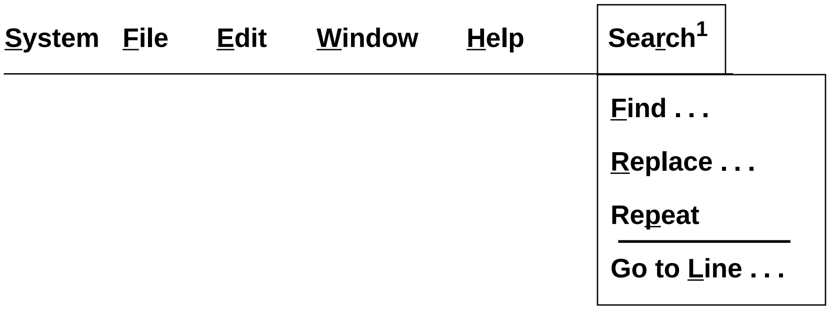


Notes:

[1] Only available when a graphics window is active.

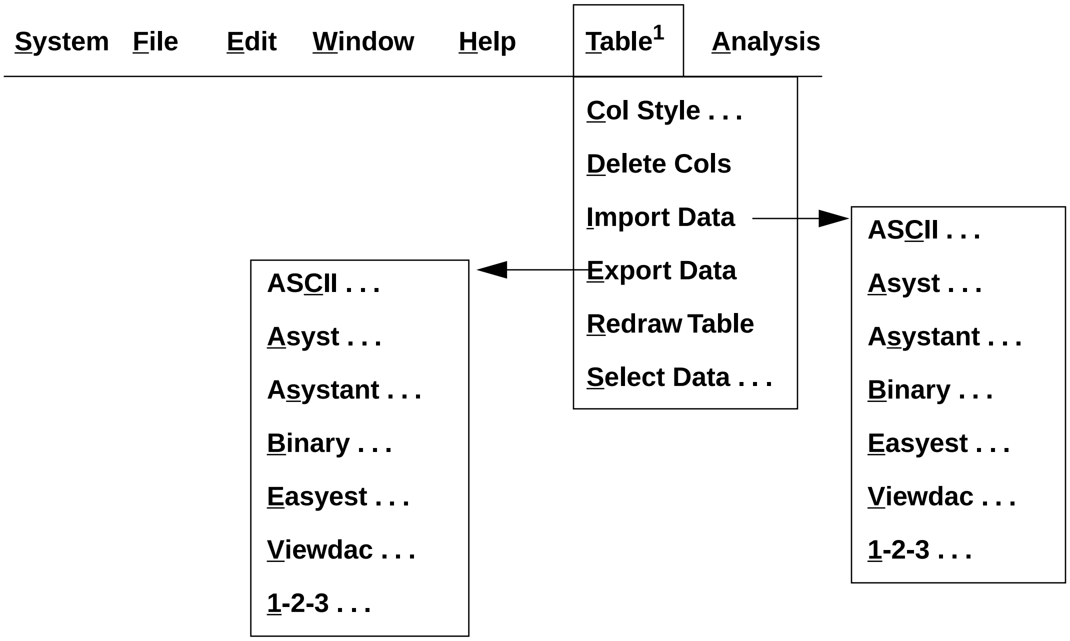
[2] Only available when a zoom window is active.

Figure B-8. Graphics Menu



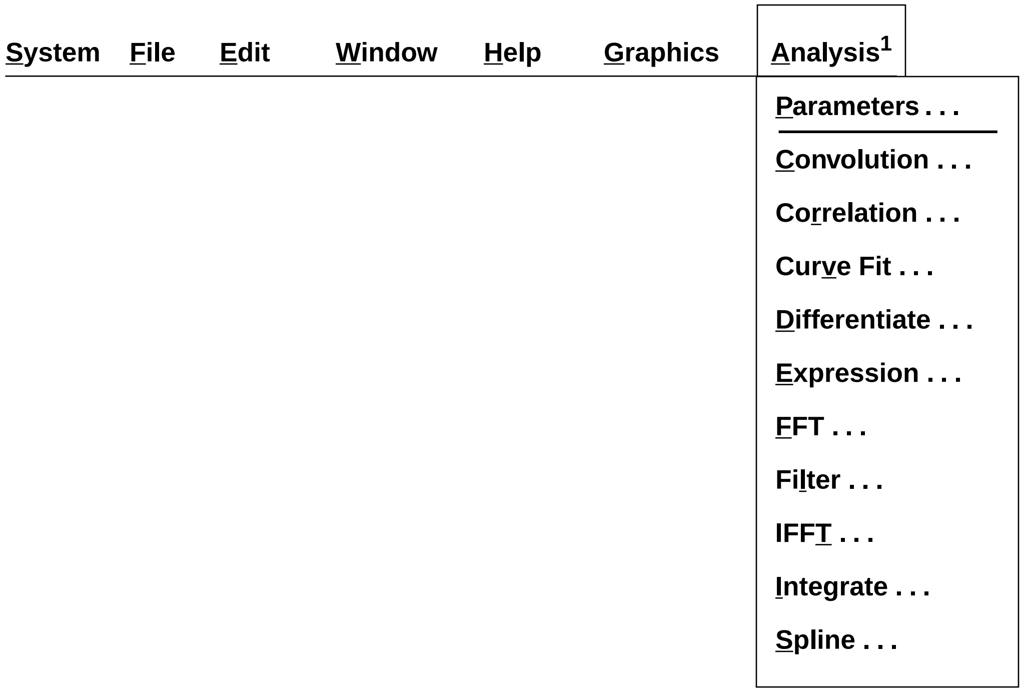
Note:
[1] Only available when a text editor window is active.

Figure B-9. Search Menu



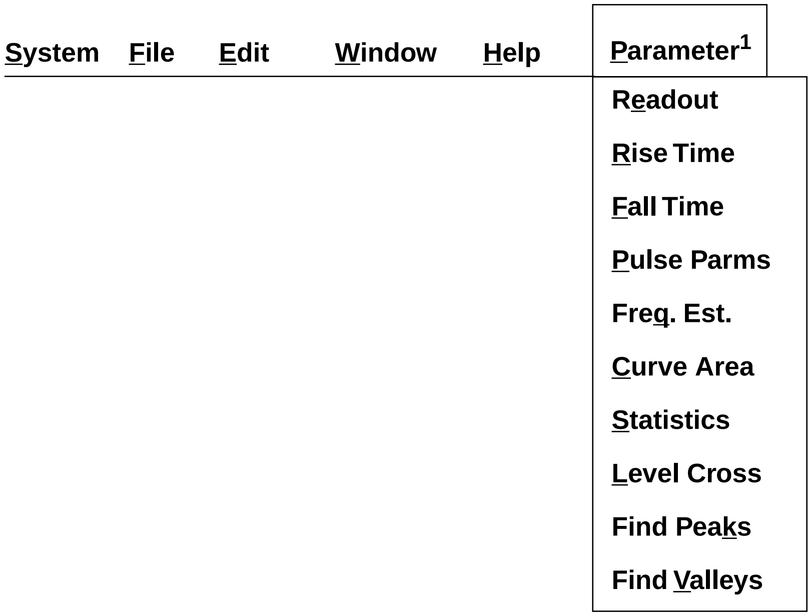
Note:
[1] Only available when a table editor window is active.

Figure B-10. Table Menu



Note:
[1] Only available when a graphics window or table editor window is active.

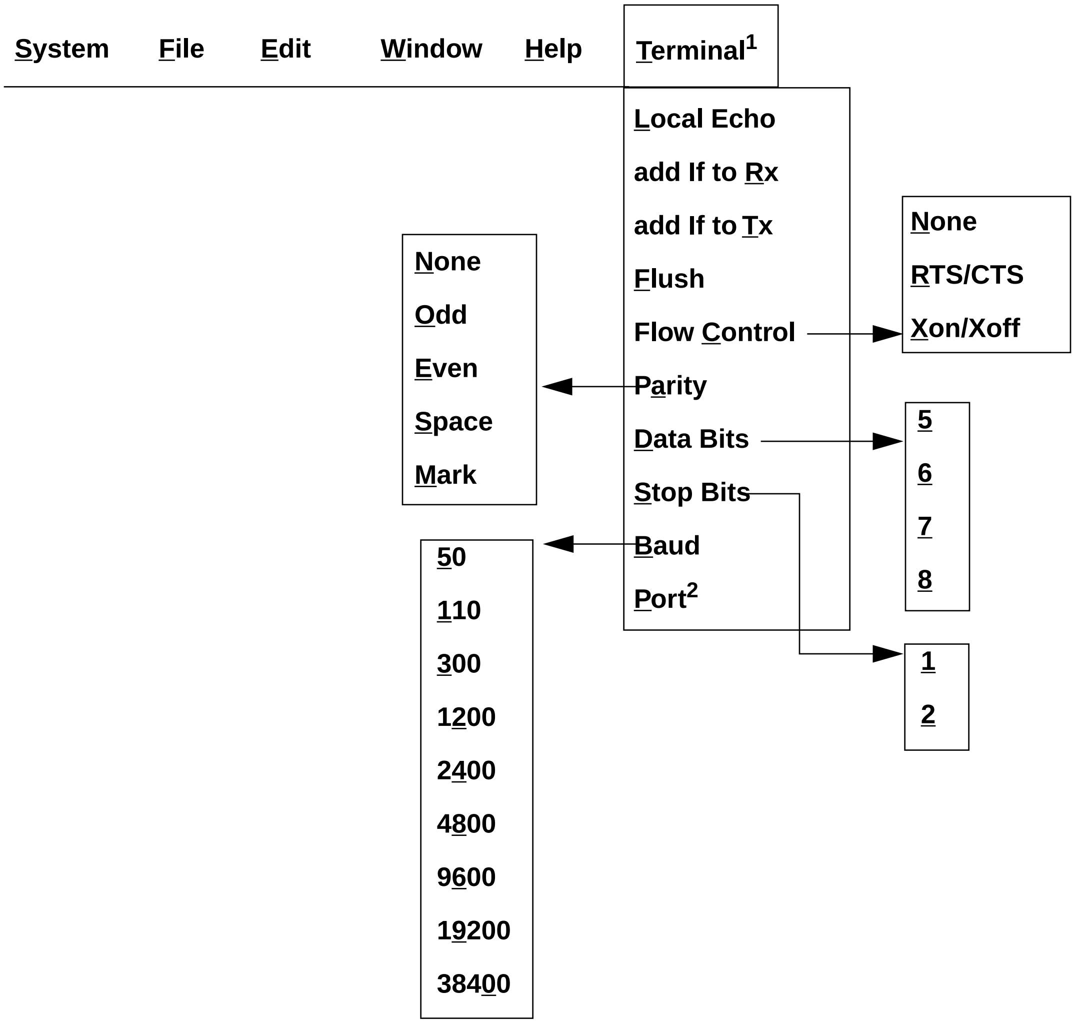
Figure B-11. Analysis Menu



Note:

[1] Only available when a parameters window is active. A check mark indicates the selected calculation routine.

Figure B-12. Parameter Menu



Notes:

[1] Only available when an RS-232 window is active.

[2] Displays a pop-up menu of available ports; a check mark indicates the selected port.

Figure B-13. Terminal Menu

C

Operators and Functions

Table C-1 contains a list of mathematical and logical operators that you can use in Expression tasks, in Analysis\Expression operations, and when performing calculations on data in table editor and graphics windows. The operators are listed in order of decreasing precedence; operators in the same box have equal precedence.

Table C-1. Mathematical and Logical Operators

Operator	Meaning
unary–	negative
~	ones complement of
**	raised to the power of
*	multiplied by
/	divided by
+	plus
–	minus
<<	shifted left by
>>	shifted right by
&	bitwise AND
^	bitwise exclusive OR
|	bitwise inclusive OR
=	equals

Table C-2 contains an alphabetical list of mathematical functions that you can use in Expression tasks, in Analysis\Expression operations, and when performing calculations on data in table editor and graphics windows. Refer to Chapter 14 of the *VIEWDAC Reference* manual for a description of these functions.

Table C-2. Mathematical Functions

ABS	EXP	SEC
ACOS	FFT_ZOOM_MAG	SECH
ACOSH	FILL	SIGN
ACOT	FREQEST	SIN
ACOTH	FROMDATE	SINH
ACSC	GRAD	SIZE
ACSCH	LN	SORT
ASEC	LOAD	SQRT
ASECH	LOG	STD
ASIN	LOGN	SUB
ASINH	MAX	TAN
ATAN	MEAN	TANH
ATANH	MEDIAN_FILTER	TC_CONV
AUTOCORR	MIN	TODATE
CLAMP	MOD	TRUNC
CONV	NRAND	TYPE
COS	POLYEVAL	UNLOAD
COSH	POLYFIT	VAR
COT	RAD	[ ]MAX
COTH	RAMP	[ ]MIN
CROSSCORR	RAND	[ ]SUM
CSC	REAL_ALLOC	^MAX
CSCH	REAL_FREE	^MIN
CSPLINE	REVERSE	^SUM
DEG	ROTATE	
DSTATS	ROUND	

D

Using VIEWRUN

This appendix describes how to create VIEWRUN run-time sequences. It addresses you, the program developer or system designer; you must provide any required end-user documentation with your distributed software. It is assumed that you understand the features of VIEWDAC before using this appendix.

You can develop and test VIEWRUN sequences in VIEWDAC using VIEWRUN emulation mode. When you are ready to distribute your VIEWRUN sequences, you can license VIEWRUN systems from Keithley MetraByte. You must have one license for each machine running a VIEWRUN system.

VIEWRUN allows users to run, but not modify, the sequences that you created with VIEWDAC. VIEWRUN includes all of VIEWDAC's analysis and graphics features.

To create a VIEWRUN sequence, perform the following steps:

1. Create a sequence in VIEWDAC.

2. Test the sequence in VIEWDAC using VIEWRUN emulation mode. Refer to the next section for more information.

3. Create a VIEWRUN configuration file. Refer to page D-4 for more information.

4. Create a VIEWRUN initialization file. Refer to page D-6 for more information.

5. Create a customized banner, if desired. Refer to page D-7 for more information.

VIEWRUN Emulation Mode

VIEWDAC (versions 1.01 and greater) can emulate the VIEWRUN environment. This allows you to develop and demonstrate possible VIEWRUN sequences; you do not have to license any VIEWRUN systems until you are ready to distribute the systems.

To use VIEWRUN emulation mode in VIEWDAC, perform the following steps:

1. Select System\Config... from the menu bar to open the System Configuration dialog box.

2. From the System Configuration dialog box, enable the VIEWRUN Emulation Active toggle button.

3. From the System Configuration dialog box, specify whether you want the menu bar to be on or off by enabling or disabling the VIEWRUN Emulation Show Menu Bar toggle button. The following sections describe the functions available for each state of the menu bar.

4. To exit VIEWRUN emulation mode when you finish testing your sequences, either disable the VIEWRUN Emulation Active toggle button in the System Configuration dialog box or press [F10]. When you exit VIEWRUN emulation mode, the menu bar is turned on.

Menu Bar On

When the menu bar is on, VIEWRUN presents the following limitations:

- VIEWRUN grays the following menu options:
  - System\Sequence Config
  - File\New\Sequence
  - Panel\Style
  - Panel\Snap To Grid

- VIEWRUN grays the names of any macros in the following dialog boxes that were loaded through the VIEWRUN initialization file:

  – Remove Macros

  – Save Macros to File

  – Play Macro

  Graying macro names prevents the user from inadvertently deleting macros used by sequences.

  Note: Any macros loaded by the user with the Help\Macros\Load menu option or created with the Help\Macros\Begin menu option are not grayed and can be played back, saved, or removed.

- VIEWRUN does not display the names of open sequence windows either in the Window menu list of open windows or in the Window List dialog box.

Menu Bar Off

When the menu bar is off, VIEWRUN presents the following limitations:

- The user accesses VIEWDAC features (such as graphics and parameters windows) using displayed front panels. Because the user cannot open sequences, at least one sequence must be opened and running when the system is started. In addition, since VIEWRUN provides no access to the File menu to open windows, certain window types, such as analysis, graphics, and table editor, must be opened by macros executed from Macro tasks in a sequence.

  Note: The macros you create for the user have full use of the menu bar (including the File\New\Sequence, System\Sequence Config..., Panel\Style, and Window\s*equence window name* options), even though the menu bar is not available to the user.

- A static text string (up to 80 characters long) is displayed in place of the menu bar. You can modify the text string to display a customized message.

- You must provide the user with a means to end a VIEWRUN session. For example, you can include a Push Button task that starts a Macro task that selects the System\Bye menu option.

- Macros run from sequences are always run at the fast speed and with no display update. (Note that when the menu bar is on, you can specify the replay speed and display update options in the Macro Settings dialog box.)

- The [F1] help key is not available. Note that if you do not intend to make the VIEWRUN help system available to the user, you can free some disk space by deleting the VIEWRUN help files (VIEWRHLP.*).

- Only the following shortcut keys are available: [F3], [Ctrl]+[F3], [F4], [Ctrl]+[F4], [F10], [Ctrl]+[C], [Ctrl]+[X], and [Ctrl]+[V]. Refer to page 2-5 for information about the shortcut keys.

Creating a VIEWRUN Configuration File

Creating a VIEWRUN configuration (CFG) file allows you to save a customized VIEWRUN environment.

Use the CFG file to specify system and printer configuration information, such as directory paths, screen patterns and colors, current printer, and window preferences. The CFG file also contains sequence configuration information, otherwise inaccessible in VIEWRUN, such as pre-emption rate, DAS buffer size, and whether or not to display messages upon loading a sequence. The CFG file is an ASCII file.

The default CFG file is called VIEWRUN.CFG. When you run VIEWRUN, the software looks for VIEWRUN.CFG in the directory containing the VIEWRUN software. To locate the configuration information in a different file and/or path, set the VIEWCFG environment variable as follows:

```
SET VIEWCFG=[drive][directory]filename.ext
```

where *[drive][directory]filename.ext* is the file containing the configuration information.

For example, to locate the configuration information in a CFG file called SPECTRUM.CFG in the directory containing the VIEWRUN software, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWCFG=SPECTRUM.CFG
```

Notes: The same environment variable (VIEWCFG) is used to locate configuration information for both VIEWDAC and VIEWRUN. If you do not use the environment variable, two CFG files (VIEWDAC.CFG and VIEWRUN.CFG) exist. The system automatically updates the appropriate CFG file (VIEWDAC.CFG if you are running VIEWDAC or VIEWRUN.CFG if you are running VIEWRUN). If you set the environment variable, you have only one CFG file; the system uses this CFG file for both VIEWDAC and VIEWRUN.

If you want to create several VIEWRUN sequences with each one using a different CFG file, you can create several batch files. In each batch file, set the VIEWCFG environment variable to a different CFG file. To run a VIEWRUN sequence, the user runs the appropriate batch file. Refer to Setting Up the VIEWDAC Environment on page 1-5 for more information about batch files.

To create a VIEWRUN configuration file, perform the following steps:

1. Enter the following at the DOS prompt to run VIEWRUN:

   ```
   VIEWRUN
   ```

2. Set up VIEWRUN the way you would like the VIEWRUN session to appear (screen colors, window preferences, and so on).

3. Select System\Save Options from the menu bar to save the current information. VIEWRUN saves the information either to the file specified by the VIEWCFG environment variable or, if no file name is specified, to the default configuration file, VIEWRUN.CFG.

Note: You can also select System\Save Options from the menu bar in VIEWDAC and then rename the configuration file created VIEWRUN.CFG.

Creating a VIEWRUN Initialization File

The VIEWRUN initialization (INI) file contains commands executed by VIEWRUN at startup.

Use the INI file to open windows, to load and run files such as macros and sequences, to specify memory parameters, to enable VIEWRUN emulation mode, to disable the menu bar and print a customized message, and to delete proprietary files after loading them. The INI file is an ASCII file that you create using any text editor.

The default INI file is called VIEWRUN.INI. When you run VIEWRUN, the software looks for VIEWRUN.INI in the directory containing the VIEWRUN software. To locate the initialization information in a different file and/or path, set the VIEWINI environment variable as follows:

```
SET VIEWINI=[drive][path]filename.ext
```

where *[drive][path]filename.ext* is the file containing the initialization information.

For example, to locate the initialization information in a file called SCOPE.INI in the VIEWRUN directory on disk drive A, enter the following either at the DOS prompt or in a batch file:

```
SET VIEWINI=A:\VIEWRUN\SCOPE.INI
```

Notes: The same environment variable (VIEWINI) is used to locate initialization information for both VIEWDAC and VIEWRUN. If you do not use the environment variable, two INI files (VIEWDAC.INI and VIEWRUN.INI) exist. The system automatically reads the appropriate INI file (VIEWDAC.INI if you are running VIEWDAC or VIEWRUN.INI if you are running VIEWRUN). If you set the environment variable, you have only one INI file; the system uses this INI file for both VIEWDAC and VIEWRUN.

If you want to create several VIEWRUN sequences with each one using a different INI file, you can create several batch files. In each batch file, set the VIEWINI environment variable to a different INI file. To run a VIEWRUN sequence, the user runs the appropriate batch file. Refer to Setting Up the VIEWDAC Environment on page 1-5 for more information about batch files.

The VIEWRUN initialization file uses the same commands as the VIEWDAC initialization file. Refer to Appendix A of the *VIEWDAC Reference* manual for a description of the INI commands and examples of how to use them. You can also refer to the sample initialization files, VRPID2.INI and VRPID3.INI, located in the VIEWRUN subdirectory.

Creating a Customized Banner

By default, VIEWRUN searches for a file called VIEWRUN.PCX and displays this file as the banner when a VIEWRUN sequence is run. (If you are running on a monochrome system, VIEWRUN automatically looks for VIEWRUNM.PCX, a monochrome version of the VIEWRUN.PCX banner.) If neither of these files exists, the VIEWRUN sequence displays a dialog box with the revision number and a start button.

If you are using an INI file, the VIEWDAC banner is not automatically displayed. However, you can use the **show.banner** initialization file command in your INI file to display the VIEWDAC banner for several seconds before the menu bar is drawn. Refer to Appendix A of the *VIEWDAC Reference* manual for more information.

You can create a customized banner, using any Paintbrush-compatible graphics package. To display your customized banner when the VIEWRUN sequence is run, rename the VIEWRUN.PCX file VIEWRUN.OLD and rename your customized PCX file VIEWRUN.PCX. If you are running on a monochrome system, rename the VIEWRUNM.PCX file VIEWRUNM.OLD and create a monochrome PCX file called VIEWRUNM.PCX.

Sample Applications

The following sections describe three VIEWRUN applications and three sample sequences that you can use for these applications. The sample VIEWRUN sequences are included on your VIEWDAC disks.

Note: You cannot access the sample sequences while in VIEWRUN emulation mode. If you are in VIEWRUN emulation mode, press [F10] to display the menu bar and to exit VIEWRUN emulation mode. The term *VIEWRUN* is used to mean either a VIEWRUN system or a VIEWDAC system running in VIEWRUN emulation mode.

Functional Workstation

In this application, several lab technicians work at separate computers acquiring data the same way (using the same sequences), but each analyzes data differently. The menu bar is available, as well as the online help feature. Sequences and macros can be loaded and run either through the VIEWRUN initialization file or through menu selection. Analysis and graphics features can be accessed either interactively (through the menu options) or through macros.

Use the VRPID1.SEQ sample sequence for a functional workstation. The VRPID1.SEQ sequence is similar to the sequence DEMO2ALL.SEQ that you created in the VIEWDAC tutorial “Build an Experiment.” It uses a PID calculation with a Moving Average task to monitor a changing setpoint value. As the sequence is running, the technician can modify the values for the setpoint, the proportional gain, the integral gain, and the derivative gain. The setpoint, PID input, and PID output values are plotted point by point on a strip chart plot. The PID input and output values are also numerically displayed. A Comparison task is used to monitor the setpoint value; if the value goes above 10 or below –10, the PID calculation is stopped.

Note that for VRPID1.SEQ, the following modifications were made to DEMO2ALL.SEQ:

- An initialization step sets variables to initial values.
- X-labels on the strip chart plot are turned off to allow the strip chart to run faster.
- The output of the PID is limited to a minimum value of –20 and a maximum value of 20.
- The PID input, output, and setpoint values are written to 50-element arrays (**pid.in.array1**, **pid.out.array1**, and **pid.set.array1**) so that the values can be used in later calculations. Once the arrays are filled, they are reset; new data overwrites the existing data.
- The current array index for the 50-element PID arrays is displayed in the front panel. The modulo 50 of the array index is calculated so that it is reset when the array is reset; the value displayed is the current array index.
- A Calculate Error Loop task has been added. This loop is started by the Go button; the first task in the loop is a Calculate Error push button. The error is defined as the difference between the normalized PID input value (**normal.input1**) and the normalized PID output value (**normal.output1**). The arrays containing the PID input and output values are normalized by subtracting their means, and then their difference is calculated. The error (**error1**) can be plotted in a graphics window.

To run the VRPID1.SEQ sequence, perform the following steps.

1. Enter the following at the DOS prompt:

   ```
   VIEWDAC
   ```

2. Select System\Config... from the menu bar to open the System Configuration dialog box.

3. Enable VIEWRUN emulation mode; keep the menu bar on.

4. Open the sequence VRPID1.SEQ; once the error has been calculated, it can be plotted in a graphics window.

Dedicated Workstation

In this application, a classroom instructor has designed a particular task for students to perform. The students are running (but not creating) sequences and performing operations using a limited feature set. Since unused VIEWDAC features might distract students from accomplishing the assignment's objective, the menu bar is not displayed. The instructor must provide front panel objects, such as push buttons and switches, so that the students can interact with the system (run macros, start other sequences, and so on). The instructor must also make menu options, such as Analysis and Graphics, available through Macro tasks.

Use the VRPID2.SEQ sample sequence for a dedicated workstation. The VRPID2.SEQ sequence builds on VRPID1.SEQ.

Note that for VRPID2.SEQ, the following modifications were made to VRPID1.SEQ:

- A macro that selects the System\Bye menu option allows the user to exit VIEWRUN. This macro is run from the Exit VIEWRUN loop; selecting the BYE push button in the Exit front panel executes the exit macro.

- A Plot Error Loop task has been added. This loop uses a macro to create a graphics window and to plot the error calculated in the Calculate Error Loop task. A Plot Error push button on the front panel (VRPID2 Panel) starts the plotting macro. A Message front panel opens when the Plot Error push button is selected and closes when the error is plotted in the graphics window. Note that if this macro is run repeatedly, multiple graphics windows are created; to avoid running out of memory, the user should close any existing graphics windows before selecting the Plot Error push button.

- An INI file, VRPID2.INI, loads macro and sequence files and sets the VIEWDAC environment to emulate VIEWRUN. The INI file command **viewrun.emulation.on** turns VIEWRUN emulation mode on. The command **menu.bar.off** turns the menu bar off. The command **menu.bar.message** displays a customized text string at the top of the screen. The command **load.macros.file** loads the macro file VRPID2.MAC when the system is started. The command **load.sequence.file** loads the sequence VRPID2.SEQ when VIEWDAC is run.

- A Reset Arrays Loop task and a Reset push button have been added to VRPID2 Panel. Selecting Reset resets the PID input and output values to ±5 of the setpoint value. The Reset push button is useful if the PID input and output values do not converge to the setpoint and changing the gains does not help. For example, if the derivative gain term is high and the setpoint changes quickly, the PID input and output values may begin to oscillate out of control; pausing the calculation and selecting Reset stabilizes the PID values.

To run VRPID2.SEQ, perform the following steps:

1. Copy all files in the VIEWRUN subdirectory into the directory containing the VIEWDAC software.
2. At the DOS prompt, enter the following:

   ```
   SET VIEWINI=VRPID2.INI
   ```

3. Enter the following:

   ```
   VIEWDAC
   ```

   Messages are displayed on the screen as VRPID2.SEQ is loaded.

OEM Workstation

In this application, a VIEWDAC sequence monitors and controls a furnace. As in the dedicated workstation, you want the user to have access to a limited VIEWDAC feature set. You design a custom start-up banner and license the sequence to glass-producing factories. Because of the sequence's proprietary nature, you implement various forms of copy protection to safeguard your sequence and its operation.

Use the VRPID3.SEQ sample sequence for an OEM workstation. The VRPID3.SEQ sequence builds on VRPID2.SEQ.

Note that for VRPID3.SEQ, the following modifications were made to VRPID2.SEQ:

- Two C programs are used for copy protection. You run the executable program SCRAMBLE.EXE; this program scrambles the sequence, macro, INI, and CFG files and names the scrambled files MYAPP#.DAT (where # is 0, 1, 2, 3). MYAPP.EXE is the executable program that the user runs; MYAPP unscrambles the scrambled files, sets up the environment variables to use the correct INI and CFG files, and starts the VIEWRUN session.

  Notes: Do not use the provided C programs for your copy protection scheme. Since these programs are available to anyone who licenses VIEWRUN, they can be "cracked." For more information about these C programs, refer to the file MYAPP.DOC included on your VIEWDAC disks.

  Other methods for copy protection include hardware protection schemes and software copy protection schemes that require the user to enter a special code.

- A batch file, MAKEAPP.BAT, is used to call SCRAMBLE.EXE for each file to be scrambled.

- The INI file command **file.dos.delete** was added to the file VRPID2.INI; this command deletes the unscrambled files once they have been loaded into the VIEWRUN sequence, preventing the user from altering or copying the developer's proprietary files.

- A CFG file, VRPID3.CFG, sets up the VIEWRUN environment; among other things, the CFG file turns off the messages displayed when a sequence is loaded.

To make and then run VRPID3.SEQ, perform the following steps:

1. Copy all files in the VIEWRUN subdirectory into the directory containing the VIEWDAC software.

2. At the DOS prompt, enter the following to scramble the VRPID3.SEQ files:

   ```
   MAKEAPP
   ```

3. Enter the following to unscramble the files, set the environment variables, and load VIEWDAC:

   ```
   MYAPP
   ```

Notes: Because of the VRPID3.CFG configuration file, messages are not displayed on the screen as VRPID3.SEQ is loaded.

Since an INI file is used with VRPID3.SEQ, a startup banner is not displayed when VIEWRUN is loaded.

Index