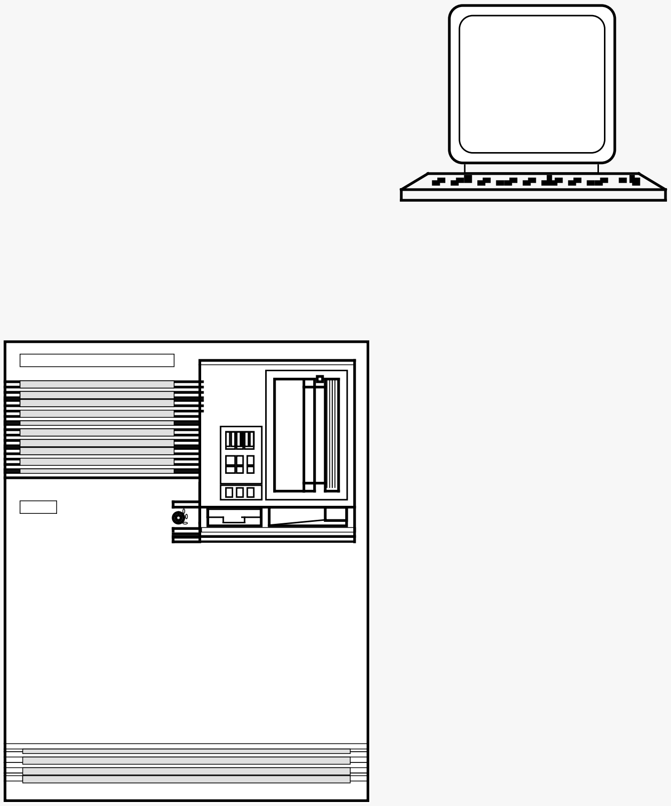
CHART/II

User's

Guide

SQUARE D COMPANY
CRISP AUTOMATION SYSTEMS



CHART/II User's Guide

Square D Company
5160 Paul G. Blazer Memorial Parkway
Dublin, Ohio 43017
USA

Dedicated to Growth
Committed to Quality

CHART/II User's Guide

Square D Company
5160 Paul G. Blazer Memorial Parkway
Dublin, Ohio 43017
USA

(614) 764-4200

Dedicated to Growth
Committed to Quality

CHART/II

User's Guide

SQUARE D COMPANY

CRISP AUTOMATION SYSTEMS

CHART/II User's Guide
Document number: 500 052-001, Rev. C

Document History

Revision	Date	Pages affected/Description of change
a	8/21/91	Preliminary
b	9/20/91	Preliminary
c	2/20/92	Preliminary

Software Version CRISP/32 Rev. 2.8 and Later

This information, furnished by Square D Company is believed to be accurate and reliable. However, Square D Company neither assumes responsibility for its use nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Square D Company. This information is subject to change without notice.

Table of Contents

Introduction

Operation

Configuring Chart/II

Display Mode

Display Creation

Display Creation (cont)

Link Switches

Display Linking

Display Directory

Initialization File

Plotting Techniques

Introduction

General

This document describes how to use the CHART/II product.

This manual is broken down into the following sections.

Section	Description
Operation *(page 3)*	This section defines the basic operational details required for CHART/II.
Configuring CHART/II *(page 7)*	This section defines the requirements when compiling and linking programs for use with CHART/II.
Display Mode *(page 9)*	This section lists all display modes and performance controls.
Display Creation *(page 13)*	This section defines all displays created with TPU or any other editor capable of generating ASCII text files.
Link Switches *(page 29)*	This section defines the switches to append to variables specified in a display file.
Display Linking *(page 33)*	This section defines the requirements when linking displays.
Display Directory *(page 35)*	This section defines the contents of the Display Directory.
Initialization File (CTPINI.CTP) *(page 37)*	This section defines the contents of the initialization file.

Section	**Description**
Plotting Techniques *(page 39)*	This section defines the various techniques available for plotting.
Appendix A - Sample Display File *(page 43)*	This appendix shows a sample display file.
Appendix B - Escape Sequence *(page 45)*	This appendix summarizes the escape sequences that may be used to control features of the terminals.
Appendix C - Terminal Setup Features *(page 47)*	This section defines the terminal attributes that should be set for CHART/II
Appendix D - Data Type Summary *(page 49)*	This section summarizes the arguments used in the plotting commands.

General

CHART/II is a CRISP/32 database display and modify software package that operates on DEC CRT terminals (and any clone terminal or PC that correctly emulates one of the DEC terminals). The following DEC terminals are currently supported.

Monochrome w/o plotting	**Monochrome w/ plotting**	**Color w/plotting**
VT100		
VT220	VT240	VT241
VT320	VT330	VT340
VT420		

CHART/II requires no special hardware or system changes and runs on any DEC VAX using a single serial port per terminal. Each terminal used for display can also be used as a normal system terminal. Terminals can also be setup to allow only CHART/II functions and not any VMS DCL operations.

The operation of CHART/II is similar to CRISP/32 color display package and provides a low cost, easy-to-use complement to the already powerful CRISP/32 display tools.

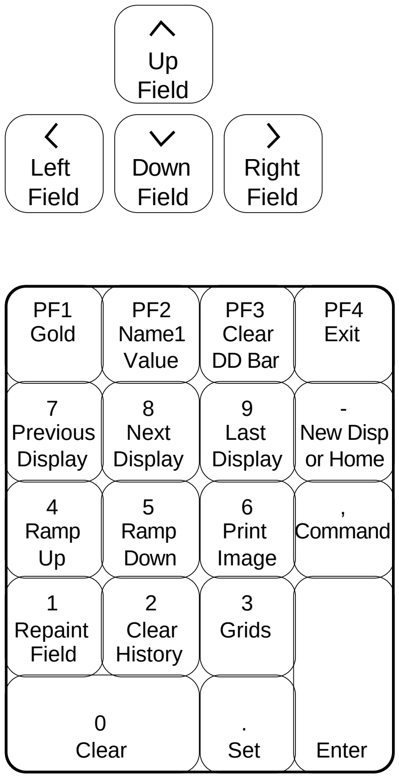
Major Features

- Simple display creation with any VMS editor.
- Accepts ANSI and DEC standard screen control escape sequences.
- Up to 600 linked fields are allowed per display screen.
- Fields are linked by database name and variable name.
- Bit variables can be displayed as any ASCII string.
- 92 Key links per display, including recipe links and F keys.
- Displays can be accessed by Name or Number or by the Previous, Next, Last and Home keys.
- Displays can be selected by a running CRISP/32 logic.
- Can be used remotely over phone lines via modems.
- Fields can be locked with a password per display.
- User selectable refresh timing and VMS priority control.
- Grid placement, size, and data scaling can be specified on a per display basis.
- Hardcopy plot output is available through the printer port of a terminal or to a queued printer.
- Up to 20 plotting grids per display.
- All plotting features can be controlled from the application program.
- Both point (scatter) plots and continuous line plots can be produced.
- Up to 20 pairs of variables can be plotted on any combination of the grids.

Major Features (cont)

CHART/II can be run on any terminal or PC capable of emulating the appropriate DEC terminal. CHART/II can also be run remotely over phone line used by plant managers from their offices or homes as an inexpensive process window.

Chart Keypad



The Left, Right, Up and Down arrow keys move to adjacent fields for VALUE and NAME changes. When prefixed with the **Gold** key (PF1), the arrow keys cause the cursor to move to the farthest link in the indicated direction.

The **Ramp Up** and **Ramp Down** keys increase/decrease a value by 1 for integers and 1% for floats.

The **Previous** and **Next** keys select the indicated displays. **Last** returns to the last screen regardless of how you got where you are now.

The **New Disp** Key operation depends on the contents of the VALUE field at the time that the **New Disp** key is pressed. If the VALUE field contains a number, that number will be used as the index of a display listed in teh CTPINI.CTP file (refer to section xxxxx). If the VALUE field contains a string, that string will be used as the name of a display. If the VALUE field is empty, the current Home display will be loaded.

The **Enter** or **Return** keys execute VALUE or NAME changes.

The **Set** and **Clear** keys will enter a 1 or 0 in the current VALUE field.

The **Clear Bar** key will clear the contents of the VALUE or NAME entry bar on line two.

The **Command** key interprets the VALUE bar contents as a command. Currently implemented commands are as follows.

- **Upassword** Unlocks the locked fields if 'password' matches the display's .PASSWORD=.
- **Usys_password** Enables special COMMANDS and the **Exit** key if 'sys_password' matches the password specified in CTPINI.CTP.
- **L** Locks all lockable fields from VALUE changes.

(Continued on next page.)

Chart Keypad (cont)

PF1 Gold	PF2 Name1 Value	PF3 Clear DD Bar	PF4 Exit
7 Previous Display	8 Next Display	9 Last Display	- New Disp or Home
4 Ramp Up	5 Ramp Down	6 Print Image	, Command
1 Repaint Field	2 Clear History	3 Grids	Enter
0 Clear		. Set	

The **Gold** key is used in conjunction with other keys to perform a function. Currently, the **Gold** key can be followed by A thru Z, a thru z, 0 thru 9, and some special characters. These keys can be linked to CRISP/32 variables. A value for the variable is also specified.

The **Clear Grids** key erases all grids on the current screen and redraws the grid lines. When the erasure is finished, all graphics will continue at the current X,Y location.

The **Repaint History** key erases and redraws all grids on the current display and also retraces the last N data samples from each graph. The N history points are replaced on each grid, left-to-right, using the current plotting pen control (line or point). Each point is replaced, left-to-right, regardless of when it was originally sampled. Note that a user history buffer must be specified for this feature to operate.

The **Print Image** key will create an image file and send it to the SYS$PRINT queue (or an alternate print queue specified by the user via the .QUEUE directive in the CTPINI.CTP file. This feature can only be used on the graphic terminals (i.e., VT240, VT241, VT330, VT340).

Operation Summary

- CHART/II can be started by entering any of the following forms of the CHT command.

  - `$CHT`

    This form of the command causes CHT to search the current directory for a file named CTPINI.CTP for use as a display control file. The first display listed in CTPINI.CTP is displayed on the screen. CTPINI.CTP is described in detail in the 'INITIALIZATION FILE' section of this manual.

  - `$CHT NAME.DSP`

    This form is the same as the one previously except that NAME.DSP is the first display put on the screen. NAME.DSP will become the new home display. Note that .DSP is the default filetype and may be omitted.

  - `$CHT n`

    This form is the same as the one previously except that the nth display listed in CTPINI.CTP is the first display put on the screen. The nth display will also become the new home display.

  - `$CHT OVERVIEW / CRISP$CH2:`

    The / allows the user to enter the device and the directory where CHART/II is to access the displays and the CTPINI.CTP file. This is useful in separating certain plant areas from other areas. The last parameter must be a valid VMS device and directory specification or a logical name whose translation is a device and directory specification. It may also include a DECnet node specification.

Operation Summary (cont)

- In general, CHART/II allows you to perform the following.
  - Use the arrows to select a field on the display. Locked fields will be skipped unless the display is unlocked.
  - Set or clear a field to 1 or 0 by using the **Set** <.> and **Clear** <0> keys.
  - Clear the VALUE bar by pressing the **Clear DD Bar** <PF3> key.
  - Select the previous, next, home or last display by pressing the **Previous Display** <7>, **Next** <8>, **Home** <->, or **Last** <9> key.
- VALUE mode allows you to perform the following.
  - Enter a new value for the current field by typing the number or string and pressing the **Enter** or **Return** key.
  - Enter a display name or number in the VALUE bar and press the **New Disp** key to bring up a new display.
  - Select the original ROOT display by pressing **New Disp** <-> with a blank VALUE bar. The ROOT is the last display requested directly by number and not by NAME, PREVIOUS, or NEXT.
  - Enter a command in the bar followed by the **Command** <,> key.

General

CHART/II is installed on a CRISP system in the directory [CRISP.CH2] and [CRISP.CH2.DEMO]. Note that the sample displays in [.DEMO] are intended as reference only. They may need to be edited to reflect your database(s) and variable names. CHART/II can be run with the sample displays provided; however, all display fields will be replaced with question marks. Refer to the Installation Guide supplied with your distribution for exact details on installing this product.

Configuration

Once CHART/II is installed, reconfigure CRISP by entering the following.

```
$ @CRISP:CRISP_CONFIG
```

When the CHART/II configuration begins, the following prompt is displayed.

```
Are you licensed for multiple terminal use of this product? [Y/N]:
```

If you are licensed for multiple terminals, configure the appropriate VAX ports for automatic startup of CHART/II.

The following prompt is displayed.

```
How many copies of this product should be automatically started:
```

Enter the number of terminals on which to run CHART/II.

If you do not want to automatically start CHART/II when your system starts, you should specify zero for the number of CHART/II copies to be automatically started.

For each copy of CHART/II that you want to start automatically the following must be entered.

- The VAX port name (e.g. TXA3:, TXA4:, LTA45:, etc.) to which the VT is connected.
- The name of the CHART/II display that you want to appear on the terminal when your process control system is started.
- The third question requests the VMS device and directory in which the displays and the CTPINI.CTP file can be located. This becomes the default directory when CHART/II runs for the specified terminal.

  The questions will follow the following format.

  ```
  Enter the terminal port [TTA0:]:
  Startup display name [DIRECTORY]:
  Display and CTPINI.CTP directory [CRISP$CH2:]:
  ```

Configuration (cont)

From this point on, CHART/II will automatically start at each terminal every time CRISP/32 is started through the use of the command file USER_START_CHT.COM, which was created during configuration. CHART/II will also exit correctly when the CRISP/32 system software is stopped via the CRSTOP command.

You can reconfigure your terminals at any time by repeating the previous procedure. You may want to become familiar with a single copy of CHART/II before setting up the automatic start at multiple terminals.

Using CHART/II

CHART/II displays are created, linked and run (displayed) by going thru the following general steps. Each step is described in more detail in the following sections.

- Use TPU to create a CHART/II display file ([CRISP.CH2.DEMO]BLANK.DSP may be used as a template).
- Place the name of the new display file in CTPINI.CTP and `DIRECTORY.DSP.`
- Start CRISP/32.
- Run EDL to link your display (this may be skipped if .AUTO_LINK is specified in the display file).

NOTE

CRISP/32 must be running before EDL.

- Invoke CHT.

General

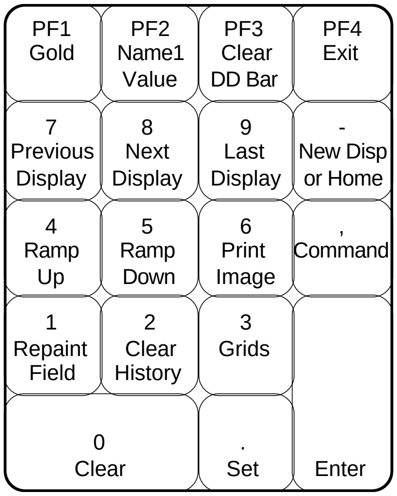
Refer to the keypad diagram for the following description.

Display Control

When CHART/II is first started, a directory page is normally displayed and you can then decide on a display to bring up. If your directory lists displays by number, then you can type the number and press the NEW DISPLAY <-> key on the numeric keypad. You can also enter the display name and press the same key on the numeric keypad.

All CHART/II displays can be referenced by name but only those listed in CTPINI.CTP (usually all of them) can be referenced by number. Your directory display should list a name, number and description for each display. It is the users responsibility to keep CTPINI.INI and DIRECTORY.DSP updated and synchronized.

Displays can also be referenced by using the PREVIOUS, NEXT, HOME (NEW DISP without an argument) and LAST display keys.



The **Previous Display** and **Next Display** keys bring up the screens specified in the current display file with .PREV= and .NEXT= (refer to the Display Creation section). The previous and next displays are shown on the top line of the CHART/II screen.

The **New Disp** key can be pressed without first entering a display name or number as described previously. In this case, the Home display listed on the top line of the screen is brought up. The Home display is always the last display requested by number or the first display to appear on the screen when CHART/II is started. This feature allows you to specify a 'root' or 'home' display, use **Previous Display** and **Next Display** to access related displays and subsequently return to the 'root' display with a single key stroke.

The **Last** key (on the numeric keypad) will bring up the display in use before the current one, regardless of how the current screen was brought up.

Cursor Control

While CHART/II is displaying a screen with unlocked fields, you can use the four cursor arrows to move from field to field. You use the left and right arrows to move across a line; and the up and down arrows to move up and down a column. The current position is indicated by a cursor under the field. Remember that all locked fields will be skipped if the display has a password and is not unlocked.

The **Gold** key (PF1) can be used in conjunction with the arrow keys to move the cursor to the extreme in any of the four directions. **Gold**-up arrow, for example, moves to the nearest field in the top line of the display. **Gold**-left arrow moves to the last field on the current line, etc.

Ramp Up/Down

The **Ramp Up** and **Ramp Down** keys change the value of the current field by +/- 1% for floats or +/- 1 for integers. The new value will be entered into the database.

Enter/Return

The **Enter** key has several uses and is normally interchangeable with the **Return** key on the main keyboard.

If you have just typed a new value to be entered into the database, the **Enter** key terminates the entry and places the value into the current field on the screen and into the database.

If you have just entered a variable name into the NAME bar, the ENTER key will cause CHART/II to search the symbol table of the current database for the CRISP/32 variable specified. See the section on mode select below.

Set/Clear

The **Set** and **Clear** keys will place a 1 or a 0, respectively, into the current screen field.

Clear Bar

The **Clear Bar** key is used to erase either the VALUE bar on the second line of the CHART/II display. This key can be used to quickly remove a mistake in the VALUE bar.

Key Link Mode

The **Gold** key is used to select the key link mode. Normally, the main keyboard keys are used to enter information into the VALUE. When the Gold mode is selected, the keys on the main keyboard take on a different function. Any key pressed in this mode is checked against the current displays key link list. Any key pressed following the **Gold** key removes the keyboard from key link mode and returns it to normal mode. Note that while in Gold mode, the word GLD appears on line two of the display.

Command Key

The COMMAND key causes the VALUE bar contents to be interpreted as a command to CHART/II. The following commands are implemented.

Field Lock and Unlock

Upassword	Unlock the locked VALUE fields.
Usys_password	Unlock for system commands.
L	Lock the lockable VALUE fields.
L	Lock against system commands.
Pnn	Set CHART/II update priority to 'nn'.

The U and L commands are normal user mode commands used to lock and unlock the current display fields. The U command requires that the user enter the password as specified in the display file for the current display.

Command Key (cont)

Performance Controls

Priority Specification

The P command can only be entered if the display has been unlocked with the system password (if any) as specified in CTPINI.CTP.

The Pnn command alters the current running priority of the CHART/II display mode to nn where nn is between 0 and 31. We recommend that CHART/II be run at 4. If you raise the priority above 4 you will find that CHART/II will run faster but the logic scan time may be unacceptably slow.

NOTE

Priorities above 15 are definitely not recommended for CHART/II

The priority is a very critical parameter that can severely affect the performance of the CHART/II displays as well as the running logic programs. The capability to change this value 'on-line' is provided only as a tool to tune your system and should not be used during normal operation. Once you determine a reasonable value for your system, you should place it in CTPINI.CTP to make it permanent.

Minimum Display Update Specification

A minimum update time in seconds can be specified per display. If none is specified, or the value is less than 1, the process will default to a 1-second update rate.

Normally, you should set the minimum update to as large a value as possible. The longer the delay, the lower the CPU load from CHART/II. Note, that the screen will refresh anytime a data value is entered and can optionally be forced to refresh early by the CRISP/32 application program (refer to .UPDATE_OVERRIDE).

Exit

The **Exit** key is used to terminate CHART/II and return to the operating system. If a system password has been specified in CTPINI.CTP, it must be entered before the **Exit** key will operate. If a terminal server port set to 'remote' is used for CHART/II, it cannot be used to log into to a VAX on the server's network.

Notes:

General

All CHART/II displays are created with TPU or any other editor capable of generating ASCII text files. This chapter discusses creation of text mode displays only. Refer to the Plotting Techniques section for information about graphical displays.

Display Creation

At run time, each CHART/II screen consists of 22 lines of display information and two lines of command and status information. The top two lines on the terminal are reserved for CHART/II use and the bottom 22 lines are for user display. This means that the first 22 lines of the display file (not counting 'directive' lines) will be placed on the lower 22 lines of the screen at display time.

When creating CHART/II displays, you can use any character available on the keyboard except braces, (i.e., { and }). The braces are used to define the starting position for a variable display field. The dot character '.' can never appear in column one of the display portion of a display file. All other characters can be used to annotate the display. Note that the <TAB> character can be used to build displays, however, you must run EDL to convert all <TAB>s to spaces before running the display. Refer to Appendix E for a sample CHART/II display file.

Blink, Bold, Reverse, Underline and Color Control

While any line of the display can contain escape sequences to control the display attributes (i.e., blink, bold, reverse and underline), normally, only text information in the display will be affected by the sequence. The live data will usually be displayed in 'normal' video. The only exceptions to this are for Logical variables when the .TRUE or .FALSE directive contains an attribute qualifier (i.e., /BLINK).

If a line contains embedded escape sequences, the characters in the sequence will be counted as though they took one space in the display. This may affect alignment of subsequent fields on the same line. It is suggested that escape sequences be placed at the end of the line preceding the line they are intended to modify. Any line starting with a period (column 1) is printed exactly as located and should not contain any display text or data fields. A line starting with a period does not count as one of the 22 available display lines but can be used to send non-printing control sequences to your terminal.

If you use escape sequences to reposition the terminal cursor, you may find that the display lines don't end up where they should be. This can be used to advantage if you understand cursor positioning. If you get lost, you can send the sequence `<ESC>[3;1H` after all other escape sequences and before the first real display line. This sequence puts the cursor at the beginning of line three where the actual display starts. If you define two lines as one double height line, they count as two display lines.

Some of the more useful escape sequences are defined in Appendix F. Others are defined in the programmer's manual for the terminal.

Field Location on a Screen

When building a display, you place a pair of squiggle brackets around a variable index number at every point on the screen that you want to display a 'live' variable from the database. Refer to the following example.

```
{123}
```

This will define a position on the display for a variable with index 123. The position is actually defined by the left '{'. The index can be any number from 1 to 32767 and is used to relate the screen position to a variable name. If you follow the {123} with only a right '}', you can define a new screen position for the variable in the database immediately following the variable identified by {123}. For example:

```
{123} } } }
```

This will display 5 consecutive variables from the database starting at the variable defined by {123}.

Variable Specification

Variables are defined at the end of the display file. The .SYMBOL command is entered into the display file on any line after 22. Variable definitions are then entered as follows:

```
.SYMBOL
INDEX/ DB:NAME/SWITCH
INDEX/ DB:NAME/SWITCH
INDEX/ DB:NAME/SWITCH
.   .
.   .
.   .
```

INDEX is the number entered in the display portion of the file as '{INDEX}'. DB is a number that specifies the database in which the variable resides. NAME is the 31 character CRISP/32 variable name to be displayed on the screen. SWITCH is an optional modifier such as /L for field lock, /K=A:1 for keylink or /-U for no update. Examples of variable definitions can be located in Appendix E.

NOTE

The index values must be entered in ascending numerical order after the.SYMBOL directive.

(Continued on next page.)

Variable Specification (cont)

The maximum number of links per display file is 600; however, the more links on a screen, the more CPU time CHART/II consumes while managing these links. String fields are significantly more CPU intensive than numeric links.

`/K=X:nnnn`	This variable is linked to key 'X'. Gold-X will assign the value nnnn to the variable.
`/L`	This variable will be locked from VALUE changes if the display is locked.
`/-U`	This variable will not be continuously updated. Default is /U.
`/T=_dev`	This variable will be set from the CRISP/32 logic and will specify the display for terminal _dev:
`/KD=`	Display function keys
`/W=`	Specify the format of the output on screen.

Variable List Specification

If you want to display a group of Logical variables from databases zero and one, the display portion of your file (lines 1 thru 22) might contain the following.

- Logicals from database zero.

```
{1} } } } } } } } } } } } } } }
```

- Logicals from database one.

```
{2} } } } } } } } } } } } } } }
```

The variable definition part of the display file (after line 22) would look like the following.

```
.SYMBOL
1/0:NEW_DB
2/1:NEW_DB
```

There are several display commands that can be entered besides .SYMBOL, and they must all follow the 22 display lines and precede the .SYMBOL command. The commands are described in detail in the following paragraphs.

Variable List Specification (cont)

You can place comments in the display files on any line after the 22 display lines but before the .SYMBOL command. Any line in this area that begins with an exclamation point ! is a comment and will be ignored. Refer to the following example.

```
.DATA_BASE = B0, B1
! B0  is a user database.
! B1 is used to calculate XYZ.
```

The two lines following .DATA_BASE are ignored by EDL during linking and CHT while displaying. They are only used to during the display file.

.AUTO_LINK

The .AUTO_LINK command temporarily links the display to the current database each time it's brought up on the screen. The feature will guarantee that the most current modifications to a display are linked even if EDL has not been run. Eventually, you should run EDL to permanently link the display.

NOTE

The .AUTO_LINK will lengthen the time it takes to bring up the display, so after the modifications are complete and the CRISP/32 database is stable, remove the .AUTO_LINK from the displays.

.COLUMN_MODE

The .COLUMN_MODE is used to control what the up and down arrow will do at the top or bottom of a column. In default mode when the cursor gets to the end of the column the cursor will wrap around to the other end of the column. .COLUMN_MODE will cause the up and down arrows to process in a vertical sequence. If the cursor is going forward (down arrow) when the cursor gets to the end of the column, the cursor will place itself at the top of next column to the right. Opposite occurs for the up arrow, going to the closest column to the left and starting at the bottom of the column.

.CURSOR=(Style Command)

Alternate cursor specification

.CURSOR= ascii_string

The text string specified can be up to 32 characters in length and will be sent to the screen immediately before the current field is printed. The command should specify the escape sequence that you want to use to modify the current field when the cursor stop there. Refer to the following example.

```
.CURSOR= <ESC>[4m
```

This command causes the field to be underlined.

.DATA_BASE=

Database specification

There are two forms of the .DATA_BASE= command. In its simplest form, the .DATA_BASE= B0,B1,B2,... command specifies the names of the CRISP/32 databases where the variables are located. These should match the database names for your system. For example, if you have two CRISP/32 databases named PROC and CONTR, the .DATA_BASE= command would look like the following.

```
.DATA_BASE= PROC,CONTR
```

The order of the database names specified here is important. The first database name will be referred to as database zero, the second as database one etc. for some operations.

The second form of the command allows you to place abbreviations for the database names on the .DATA_BASE= command line. The abbreviation must be defined in CTPINI.CTP. For example, if your actual database names were PROCESS, CONTROL, and TEST, you could define databases 0, 1, and 2 in your display file as follows.

```
.DATA_BASE= PR, CN, TS
```

You must then define the abbreviations PR, CN, and TS in CTPINI.CTP with the following lines.

```
.DATA_BASE PR=PROCESS
.DATA_BASE TS=TEST
.DATA_BASE CN=CONTROL
```

This feature allows you to quickly change the symbol table assignment in CTPINI.CTP without changing every display. Note, that if the abbreviation is not defined in CTPINI.CTP, the database must be located by the name specified in the .DATA_BASE= command line of the display.

You can, therefore, mix abbreviations and actual symbol file names in the display file if you think you can keep it all straight.

.Gn=

Grid size and control definition

```
.Gn= HC, WC, HL, WL, GC
```

Where HC and WC are the desired height and width of the grid in characters. HL and WL are the number of horizontal grid lines and the number of vertical grid lines. HL=1 and WL=1 will draw a single x axis and a single y axis. GC is the index to a numeric variable that specifies the type of grid being drawn. Currently, the following grid codes are implemented.

Value	Description
0	Normal grid. No erasure bar
1	Vertical erasure bar precedes the x value of the first .XY for this grid.
2	Bordered grid. No erarure bar.
3	Bordered grid. With erasure bar.

.Gn= (cont)

The erasure bar is a vertical section of the grid that is redrawn immediately in front of the line being drawn. The line would normally be drawn continuously and could end up over old data without the erasure bar. The erasure bar should not be used with point plots.

The grid border caused a rectangle to be drawn outside the grid area.

If more than one X,Y pair is specified on the same grid, the erasure bar will precede the first pair defined. For the best results with an erasure bar, all Y values on the grid should use the same X value. The following is an example of the grid command that will draw a grid 10 characters high and 20 characters wide.

```
.G3= 10, 20, 11, 21, 999
```

There will be a grid line at 0, 0 (the origin) and between each character thereafter, including a line after the last vertical and horizontal character. The numeric variable specified by index 999 (refer to .symbol) contains a code to control the grid.

A value of zero for HL and/or WL allows an origin to be defined for the grid but does not produce any grid lines. This phantom grid can be used as usual to plot X,Y pairs or it could be used to erase a selected area of the screen.

Each grid (up to eight) must have a .G definition. You can then place X,Y data pairs on any or all defined grids by defining the X, Y data pairs as described in .XY pair definition.

.IDENTIFY=

CHART/II Current Display Monitor

The .IDENTIFY command specifies a CRISP Numeric variable where CHART/II can return the current display number. The form of the command is as follows.

```
.IDENTIFY = cv index
```

When a display is first brought up by CHART/II, its display number will be placed in the crisp variable relating to the CRISP variable index (cv index). The CRISP variable will be set after the screen has updated once. If the display is brought up by name, a -1 will be placed in cv.

NOTE

1. **The 'cv index' is the variable index used in the symbol table.**

2. **The CRISP variable should be unique for each terminal that CHART/II is running on. If the variable is not unique the value of the screen will give a false reading.**

.MIN_UPDATE=

Minimum Display Update command

The .MIN_UPDATE command allows the user to specify how often the display is updated. This command is per display. A typical command would look like the following.

```
.MIN_UPDATE = 4.5
```

This would tell the display to update every 4.5 seconds. This gives the user the advantage of tuning each display individually.

If the update takes less than 4.5 seconds, there will be a delay for the difference plus an additional .5 second. This allows for some delay no matter how long an update actually takes.

.NEXT= / .PREVIOUS=

NEXT and PREVIOUS display specification

The .PREV = name and .NEXT = name are used to specify displays, by name, that will be brought up at run time when the PREVIOUS and NEXT keys are pressed while the current display is on the screen. For example:

```
.PREV = DIRECTORY
.NEXT = HELP
```

The file names specified can be any CHART/II display file and the .DSP is assumed if not entered.

.NONAME

During normal operation, the variable name for the current cursor position is displayed on line two of the display next to NAME. The time taken to update this field is negligible at 9600 bps but can slow the progress of the cursor when running over phone line at 300 or 1200 bps.

The .NONAME command inhibits the variable name display. If you wish to know the name of the variable for a specific field, you can force the name to appear by entering the NAME mode.

.NO_PARK

The CHART/II product will normally place the cursor in a 'hidden' position known as the 'PARK' position. This spot is logically the link-0 position, and the user may move the cursor to the first 'real' link by pushing any of the arrow keys. The cursor will 'return' to this PARK position when the cursor is on the first 'real' link and the user pushes the left arrow. Some applications call for the display to present the cursor on the first 'real' link. When this directive is used, the cursor will ignore the PARK position, and consequently appear on the first unlocked user link.

.PASSWORD= (In Displays)

Display password specification

The .PASSWORD= command is used to specify a 6 character display password. If the command is not specified, the display cannot be locked. If a blank password is specified, the display can be unlocked with the 'U' command. If 'XYZ' is the password then the display can be unlocked with 'UXYZ'. When a display is locked, all fields with the /L switch will be locked from VALUE mode changes and the VALUE mode cursor will not stop on the field. A typical command might look like the following.

```
.PASSWORD= CHTSYS
```

.PRINT_SCREEN=

This directive specifies a CRISP variable that is to be used to control generation of a screen hard copy from the current CHART/II screen. This feature can only be used with the graphic terminals (VT240, VT241, VT330, VT340). The proper format is as follows.

```
.PRINT_SCEEN= ccode, fname
```

`ccode` is a CRISP variable that must be numeric. The variable sent from CRISP to CHART/II will tell CHART/II to make an image copy. If the code is 1, only the image file will be created. If the code is 2, the image file will be created and then sent out to the user-designated print queue. If the code returned is not 0,1, or 2 the value will be ignored and will be cleared.

`fname` is the file name of the image file created. By using a CRISP string for the file name, the file can have a unique file name, representing a process complete, a shift change, a batch number, or any other state necessary. The file can also be sent across the Ethernet network to another system for storage. A pixel image file is created from the terminal and can be stored or printed. The file is in SIXEL-1 format compatible with the LA50 and LA75 bit-map protocol.

.PRINT_SCREEN= (cont)

When activated, the background screen color of the DEC terminal will change to black. A header will appear at the top of the screen giving the name of the display, name of the image file, and the time the image file was made. The screen will stay this way for a time period of 2 to 5 minutes. The creation of the image file is complete when the user's display resumes updating. The VT300 series terminals will return to the original color, also.

.STRING_REFRESH=

String Refresh Timing. The .STRING_REFRESH = n command is used to control screen refresh for string variables. Strings are normally of less importance than other database variables and are also more time consuming to display. You can use this command to control how often the string variables are updated relative to other variables on the screen. The argument 'n' specified with this command is the number of complete screen updates between string variable updates. If you have string variables that never change, you should specify the /-U no-update switch for the variable. A typical command of this type might appear as:

```
.STRING_REFRESH= 10
```

Normally the VALUE mode cursor consists of a reverse video block around the current field. You can override this on a per display by using the following command:

.TRUE= / .FALSE=

Bit Variable Display Control. Each display file can define its own text strings to be displayed for each state of a CRISP/32 logical boolean variable. The defaults are '1' for the true state and '0' for the false state. The example below will cause the word 'ALARM' to be displayed if the CRISP/32 bit variable is a one. The word 'OK' will be displayed if it is a zero.

The text can be any character string up to 32 characters. While escape sequences are NOT recommended for use here, they will be treated like any other character and can generate interesting and sometimes useful affects. An optional syntax is available to force the TRUE string specified by the user to blink. The first example illustrates the original syntax, and the second example shows the newer option.

```
.TRUE = ALARM
.FALSE = OK
.TRUE ="ALARM"/blink
.FALSE =" OK"
```

If you choose to leave one of the states blank (e.g., .FALSE =), CHART/II will oblige by clearing the field when the variable is in that state. You will also find that the cursor will disappear when you move to the blank field. If you want a blank field with a cursor, you can specify the state as follows.

```
.FALSE = <ESC>[m
```

The number of spaces between the '=' and the <ESC> determines the length of the VALUE cursor when displaying a blank field.

.UPDATE_OVERRIDE=

The update override is a CRISP Logical variable that when set true will override the MIN_UPDATE timer, causing the display to refresh at that point. When CHART/II senses the CRISP variable is true, the variable will be reset back to false. The CRISP variable MUST be a Logical.

The following shows a command line format.

.UPDATE_OVERRIDE = (CRISP Logical variable index number)

The following shows an example.

```
.UPDATE_OVERRIDE = 34
.SYMBOL
34/0:SCREEN_OVERRIDE:B:3276
```

.XY=

X, Y pair definition

```
.XY= GRID, X_INDEX, Y_INDEX, PEN_INDEX, CHART_DONE,
DATA_CONTROL_INDEX, START_OF_HISTORY_BUFFER,
LENGTH_OF_HISTORY_BUFFER, X_EGU_AT_0,
Y_EGU_AT_0,X_EGU_PER_CHAR, Y_EGU_PER_CHAR
```

Where GRID is the number (0-9) of the grid on which the X, Y pair is to be plotted. Note that several X, Y pairs can be placed on the same grid.

X_INDEX and Y_INDEX are the index numbers of the variables to be plotted. Refer to the .SYMBOL command in the Display Creation section for a description of index numbers.

PEN_INDEX is the index of a variable used to control the simulated pen as defined in the following pen control codes.

Pen value	Description
-102	History repaint request.
-101	Future.
-100	Clear the Grid.
-16	Use line color M4 (VT340 only)
-15	Use line color M15 (VT340 only)
-14	Use line color M14 (VT340 only)
-13	Use line color M13 (VT340 only)
-12	Use line color M12 (VT340 only)
-11	Use line color M11 (VT340 only)
-10	Use line color M10 (VT340 only)
-9	Use line color M9 (VT340 only)
-8	Use line color M8 (VT340 only)
-7	Use line color M7 (VT340 only)
-6	Use line color M6 (VT340 only)
-5	Use line color M5 (VT340 only)
-4	Use line color M1 BLINK
-3	Use line color M2 BOLD
-2	Use line color M3 FG
-1	Plot with pen up but still update history buffer if available.
0	Disable plotting. No history update.

.XY= (cont)

Pen value	Description
1	" . "
2	" + "
3	" X "
4	" 0 " with " . "
5	Diamond solid.
6	Box with " . "
7	TRIANGLE solid.
8	Histogram bars (VT340 only)
9	Line plot fill (VT340 only)
10	" 0 " solid.
11 to 20	Same as 1 thru 10, with connecting line (M3 color).
21 to 30	Same as 1 thur 10, with connecting line (M2 color).
31 to 40	Same as 1 thru 10, with connecting line (M1 color).
41 to 50	Same as 1 thru 10, with connecting line (M4 color).
51 to 60	Same as 1 thru 10, with connecting line (M5 color).
61 to 70	Same as 1 thru 10, with connecting line (M6 color).
71 to 80	Same as 1 thru 10, with connecting line (M7 color)
81 to 90	Same as 1 thru 10, with connecting line (M8 color)
91 to 100	Same as 1 thru 10, with connecting line (M9 color).
101 to 110	Same as 1 thru 10, with connecting line (M10 color).
111 to 120	Same as 1 thru 10, with connecting line (M11 color).
121 to 130	Same as 1 thru 10, with connecting line (M12 color).
131 to 140	Same as 1 thru 10, with connecting line (M13 color).
141 to 150	Same as 1 thru 10, with connecting line (M14 color).
151 to 159	Same as 1 thru 10, with connecting line (M15 color).

The Pen controls 8 and 9 are for Histogram and line plot filling type displays. Pen control 8 (8,18,28...158) produces a histogram type plot. The bars are a character in width and are centered on the data point. As the bars move they will overlap themselves if necessary. When histograms are plotted in conjunction with the history repaint, the bar width is calculated according to the number of points in the historical buffer. The width is calculated to the nearest pixel. If the border is smaller than one pixel, the bars will overlap at run time.

The Pen control 9 (9, 19, 29...158) produces a line plot filled to the bottom of the grid. This is used in conjunction with pen controls -2 thru -15.

NOTE

Sometimes in the construction of the fill regions, the points will conflict end there will be a random line drawn across the fill area. This is known problem in the VT340.

Once you have defined the GRID location and sizes, and the X,Y variable pairs, you can control the appearance of each graph from the application program by changing the PEN variable as required.

CHART_DONE
DATA_CONTROL_INDEX
START_OF_HISTORY_BUFFER
LENGTH_OF_HISTORY_BUFFER

All four of the previous entries are indexes to database variables.

CHART_DONE returns to the user program the pen code just executed. This can be ignored or used to synchronize the operation of the user program with the Chart pen and grid operations.

The DATA_CONTROL_INDEX points to a database variable that can be used to control data storage and history repaint operations. The following codes are added together and placed in the data control variable by the applications program. The defaults for codes not specified are also described in the following.

Code **Description**

1 Chart/II uses the user specified data area for HISTORY DATA point STORAGE. Data buffer length must be greater than zero.

Default: If this code is not specified, NO PLOT

HISTORY is saved. The history repaint still functions if you specify a non-zero buffer length. You can then load the buffer from some other source.

2 This code causes Chart/II to AUTOmatically LIFT the PEN when the X value changes directions.

Default: If not specified the pen will remain at the user selected value.

4 If code 1 is specified, this code indicates that the user specified history buffer starting point is the "LAST" of N history points.

The history data will be written and restored from this point backwards in the database.

Default: If code 2 is not specified, then the history start is the "FIRST" of N history points.

8 Values are placed in the history buffer as located. Old data between new Xs is left alone. Normally use this code for point scatter plots where new Y data can appear in any X order.

Default: Places -9999s in the history buffer between non-consecutive X values. This has the effect of erasing old data from the history buffer between new points if a new Y value is not calculated to fall in every slot. Use this code when you are plotting history with lines.

START_OF_HISTORY_BUFFER specifies the starting point in the database where the plotting history is to be saved and restored. The direction code in the data control variable indicates whether this is to be the first or last point in the history table. LENGTH OF HISTORY BUFFER points to a variable that specifies the number of history points to be stored or repainted. A value of zero indicates
that there is no history buffer.

X EGU AT 0 and Y EGU AT 0 are the actual user's engineering units at the grid origin. These are actual values and not indexes. These values are used to scale the user's data at run time.

X_EGU_PER_CHAR and Y_EGU_PER_CHAR specify the user's engineering units per character in the X and Y direction. These are actual values and not indexes. These values are used to scale the user's data at run time.

If the user's data was expected to cover from 50 to 100 degrees C in the X direction and 100 to 200 PSI in the Y direction with a grid 10 characters high and 10 wide, the scaling information might be as follows.

SCALE FACTOR	VALUE	DESCTIPTION
X_EGU_AT_0	50	Degrees C at the origin.
Y_EGU_AT_0	100	PSI at the origin.
X_EGU_PER_CHAR	5	Degrees C per character in X Total X span is 5 * 10 = 50 Degrees C.
Y_EGU_PER_CHAR	10	PSI per character in Y. Total Y span is 10*10=100PSI

In this example, the X axis is 10 characters wide and each character is worth 5 degrees. The X axis can, therefore, handle X values from 50 degrees C, at the origin, to 50+(5*10)=100 degrees at the far right.

The Y axis is 10 characters high and each character is worth 10 PSI. The Y axis can, therefore, handle Y values from 100 PSI, at the origin, to 100+(10*10)=200 PSI at the top of the grid.

NOTE

Chart/II makes no attempt to keep the user data within a grid area or even on the screen.

In the previous example the .G and .XY commands are as follows.

```
.GO = 10, 10, 11, 11, 999

.XY = 0,1, 2, 3, 11, 12, 13, 14, 50, 100, 5, 10

.SYMBOL

1/0 : DEGREES

2/0 : PSI

3/0 : PENO

11/0 : DONE

12/0 : CONTROL

13/0 : BUFFER

14/0 : BUFFER.LENGTH

999/0 : GRID. CONTROL
```

The .SYMBOL command is used to define the required variables.

Display Creation Summary

1. Use EDT or TPU to create new displays by any name you please. .DSP should be used for the extension. Lines 1 thru 22 are display information and lines 22 and up are control information for the display.

2. Place the name of the new display file in CTPINI.CTP. The names are entered, one per line. The first name is display #1, etc. The display file named DIRECTORY.DSP should be used to list the displays and their description. This is for your information only.

Display Creation Summary (cont)

3. Specify the database names being used by placing .DATA_BASE= NAME1, NAME2,... on any line after line 22 of the display but before .SYMBOL. Abbreviations for database names can be placed in CTPINI.CTP.

4. Variables from the CRISP/32 databases are placed anywhere in the first 22 lines of the display by placing an ID number between left and right braces as follows: `{INDEX}`. These are the only characters that can't be used to annotate a display. Consecutive right squiggle braces will display consecutive variables from the database at that spot in the display. The ID number references the variable name list following .SYMBOL which can be on any line after line 23 of the display. One symbol per line is entered in the form ID/N:NAME:T:OFFSET/SWITCH. Where ID is < 32767, N is the database number, and NAME is your 30 character CRISP/32 variable name. If only ID/N:NAME is provided, you can run the EDL program to calculate the data type T and OFFSET.

Notes:

Link Switches

General

CHART/II currently allows the following switches to be appended to variables specified in a display file.

`/K=X:nnnn` This variable is linked to key X. **Gold**-X will assign the value nnnn to the variable.

`/W=sss.dd` Field format control.

`/KD=` Key-to-display link control.

`/L` This variable will be locked from VALUE changes if the display is locked.

`/-U` This variable will not be updated continuously. The default is /U.

`/T=_dev:` This variable will be set from the CRISP/32 logic and will specify the display number for _dev:. The device specified must be a VMS terminal port that has been set up in CTP_REMOTE.CFG.

/K Key Link Switch

The /K switch allows you to link a CRISP/32 database variable to a key on the main keyboard. The key letter is specified after the '=' and can be upper or lower case alphabetic, numeric, or one of the special characters on the main keyboard. The key name can be followed by a colon and value. When the Gold mode is selected at run time, the named key can be pressed to assign the value nnnn to the variable. Integer, Float, Logical and String variables can be linked in this fashion. Additionally, the /K switch allows some of the LK201 F and E keys to be linked directly to CRISP variables in the same manner that the alpha-numeric keys are linked.

For example:

`/K=B:1`	Pressing **Gold**-B places a 1 in the variable.
`/K=E2:1`	Pressing E2 places a 1 in the variable.
`/K=F9:10`	Pressing F9 places a 10 in the variable.

Note that for strings, only four characters can be assigned per key. Also note that the variable to which the value is linked does not actually have to appear on the screen and that the same key can be linked to several variables. These later features allow you to enter one of several multi-variable recipes with a single key stroke. Refer to the following example.

/K Key Link Switch (cont)

- Key linking a recipe

```
.SYMBOL
1/0:SETPOINT1/K=A:100
2/0:SETPOINT2/K=A:100
3/0:SETPOINT3/K=A:100
4/0:SETPOINT4/K=A:100
40/0:RECIPE.NAME/K=A:PRDA
41/0:START.PBA/K=A:1
42/0:START.PBB/K=A:0
43/0:START.PBC/K=A:0
5/0:SETPOINT1/K=B:200
6/0:SETPOINT2/K=B:200
7/0:SETPOINT3/K=B:200
8/0:SETPOINT4/K=B:200
80/0:RECIPE.NAME/K=B:PRDB
81/0:START.PBA/K=B:0
82/0:START.PBB/K=B:1
83/0:START.PBC/K=B:0
9/0:SETPOINT1/K=C:300
10/0:SETPOINT2/K=C:300
11/0:SETPOINT3/K=C:300
12/0:SETPOINT4/K=C:300
120/0:RECIPE.NAME/K=C:PRDC
121/0:START.PBA/K=C:0
122/0:START.PBB/K=C:0
123/0:START.PBC/K=C:1
```

In the previous example, the key A will assign 100 to four different setpoints. The key B will assign 200 to the same setpoints and the key C will assign 300. The same keys also assign either 'PRDA', 'PRDB' or 'PRDC' to the CRISP/32 string variable RECIPE.NAME and set one of three start pushbuttons while clearing the other two.

/W Width Field Control Switch

The /W switch allows the formatting of the data field. The sss portion is the overall size of the data field. This includes the decimal point. The dd is the number of digits to the right of the decimal point. Refer to the following example.

```
134/0:OVERFLOW_VALUE:F:089654 /W=12.4
```

The resulting field would be as follows.

```
#######.#### or 1234763.3456
```

If a zero is placed in the decimal field (dd), no decimal point will show up in the format.

/KD Key Display Switch

The /KD switch allows keys F7-F20 and E1-E6 to be linked to a display number when the key is pressed. Chart will bring up the display number associated with the key and the display number will also be placed in the variable specified. Refer to the following examples.

`/KD=F17:25`	When F17 is pressed, display 25 is presented.
`/KD=E2:14`	When E2 (the Find key) is pressed, display 14 is presented.

NOTE

In some cases F10 will react as a control Z. To prevent this the following steps are suggested when using a VT330 - VT340 series terminal.

1. **Enter the setup menu of the VT by pressing <F3>.**
2. **Place the cursor on the 'User-Defined Key SET-UP' and press the <ENTER> key.**
3. **Place cursor on selection 'Clear All Keys', then press the <ENTER> key.**
4. **Press the <SELECT> key. This will take you back to SETUP DIRECTORY.**
5. **Place the cursor on 'General SetUp' and press the <ENTER> key.**
6. **Place the cursor on 'Lock User-Defined Keys'.**
7. **Press the <ENTER> key until the field next to 'Lock User-Defined Keys', says 'Locked'.**
8. **Press the setup key <F3> and return back to DCL Level.**

The terminal is now setup to not allow F10 to react as a control-Z.

/L Lock Field Switch

The /L switch can be appended to any variable specification in a display file to indicate that the user should not be able to alter the field when the display is locked. You must specify a .PASSWORD=password in the display file to actually lock the display. If the password is then entered at display time via the U command, all /L fields will be unlocked and thus modifiable.

/-U No Update Switch

The /-U switch is used to tell CHART/II that a particular field is really a constant that doesn't change. CHART/II will then skip the update and associated calculations for that field. This is also useful for strings that rarely, if ever, change. The /-U switch should be used, where possible, to speed up display refresh.

/T Physical Terminal Switch

`/T=_dev:` Display drive up from CRISP/32 logic. The /T switch is used to link a CRISP/32 NUMERIC variable to a CHART/II terminal (i.e. _dev:). Each terminal should have its own CRISP/32 variable for display control. When CHART/II brings up a display and encounters a variable with the /T switch, it checks to see if _dev: matches its own startup terminal name (e.g. _TXA7:) (refer to the Starting CHART/II section). If there is a match, then that variable becomes the display driveup for that terminal until another /T match is located in this or subsequent displays. (Note that _dev: must be a legal VMS port name when used with the /T switch. All /T switches that do not match the current terminal number or that are linked to CRISP/32 variables other than type NUMERIC are ignored.

If the /T switch is used without a terminal name, the current startup terminal name will be assumed and the associated variable must be used for display control.

While CHART/II is running, the display control variable (if any) is checked once per display update pass. If the value of the variable from the database is a legal display number for the running terminal, that display will be selected. CHART/II will then clear the variable in the database and wait for it to change back to a display number.

Be sure that you use an edge-triggered CRISP/32 SET; statement to load the display number. A continuous LET; statement would reset the display number after CHART/II has cleared it and the same display will be continually redrawn.

All of this allows different variables to drive up different displays from different CHART/II terminals.

In its simplest form, you should place all /T switches in one display that is always brought up first on all terminals (e.g., DIRECTORY.DSP). Referring again to the section on starting CHART/II, you could link the CRISP/32 numeric variables CRT2 and CRT3 to terminals TXA2: and TXA3: as follows:

```
.SYMBOL

100/0:CRT2/T=_TXA2:/K=A:10
200/0:CRT3/T=_TXA3:/K=A:20
```

In this example, the value is not actually being SET; by the CRISP/32 logic but rather by a CHART/II key link to the letter 'A'. If you are at TXA2: running CHART/II and press Gold-A, the CRISP/32 numeric CRT2 will be set to 10 and display 10 will be brought up on your screen. Note that if CHART/II is running at TXA3:, display 20 will also appear there.

General

You can link (or relink) your displays using the CHART/II display linker EDL. You must relink any display that has variables that have been moved in the database since the last link unless the display file contains an .AUTO_LINK directive.

If, for example, a display is all CRISP/32 Numeric variables and the CRISP/32 database is modified changing the order or location of the Numerics, you will have to relink the display. If, on the other hand, you simply add Numerics to the database and don't change the location or order of the Numerics used in the display, you do not have to relink the display. It is a good practice to only add variables to the database at the end of a particular data type's declarations. If you never change the order of items in the database and never insert new items before old items of a particular data type, you will never have to relink your displays. This assumes, of course, that you don't add the new variables to an old display. Anytime you add variables to a display file, the file should be relinked.

Display Linking

Every new display must, of course, be linked. The display linker adds information to your display for each variable listed after the .SYMBOL command. You can still add and delete variables and switches but should not remove the information added by EDL unless you plan on relinking. EDL also converts all <TAB> characters, located in the display portion of each file, to spaces. Therefore it is possible that a display will not appear correct at run time if it has not been linked by EDL. A typical variable in a display file, before linking, might look like this:

```
.SYMBOL
1/0:HILIMIT/K=A:100/L
```

After linking the symbol would appear as follows:

```
.SYMBOL
1 /0:HILIMIT:F:36/K=A:100/L
```

The EDL program has provided the data type 'F' for Float and the database offset of 36. If you choose to edit the switches on this variable, there will be no need to relink. If, however, you change the name or remove the data type or offset, you will have relink the display.

How to Link a Display

EDL requires that CRISP/32 be running and that all databases referenced by your displays be 'installed'.

The EDL program can be run in one of the following two ways.

`$EDL NAME.DSP`	To link a single display file.
`$EDL CTPINI.CTP/`	To link every display in CTPINI.CTP. The trailing slash (/) is required.

(Continued on next page.)

How to Link a Display (cont)

If you have a shorter list of displays to link, you can place their names (one per line) in a file using EDT and then link them as follows.

```
   $EDL DISPLAY.LST/
```

All display files listed in DISPLAY.LST will be linked, one at a time.

Examples

Single display are linked as follows.

```
   $EDL NAME.DSP        ! Link CHART/II display
                        ! NAME.DSP
   $EDL -- V8.1         ! EDL prints its version.
   $EDL -- Completed    ! Your display is linked.
   $ ..
   $ ..
```

All displays in CTPINI.CTP may be linked as follows.

```
   $EDL CTPINI.CTP/     ! Link every display.
   $EDL -- V8.1         ! EDL prints its version.
   $EDL -- Completed    ! Your display is linked.
   $ ..
   $ ..
```

Display Directory

General

All CHART/II displays can be referenced by either name or number. Displays that you want to reference by number must be listed in order, one per line, in CTPINI.CTP. Displays that you want to reference by name should also be placed in CTPINI.CTP, but this is not required. Any display file in the current directory can be referenced by its file name. Since displays can be controlled from your CRISP/32 logic by display number, you must place all displays in CTPINI.CTP that you intend to control from CRISP/32 logic. (Refer to the /T switch description.)

The display listed in CTPINI.CTP should be devoted to a display directory listing. Again, this is not a requirement but just a suggested practice to keep track of your displays. When CHART/II is started (without a specific display file named), the first display in CTPINI.CTP will be placed on the screen.

The CTPINI.CTP file also contains command information in the first few lines and, therefore, must be present regardless of whether or not it is used to list displays.

⚠ CAUTION

CHART/II will not run without a local copy of CTPINI.CTP.

Notes:

General

As described in the previous section, CHART/II requires a file by the name of CTPINI.CTP before it will run. The first few lines of the file contain system-wide CHART/II directives and each line begins with a period (.). All display names are then listed following the last directive.

Currently, the following directives are implemented:

- .Customer_Name>
- .Customer_Location>
- .Customer_SW_License>
- .Customer_Use_Limit>
- .Customer_Config_Code>
- .Customer_Config_ID>
- .Data_Base
- .High_Pri=
- .Low_Pri=
- .Password=
- .Queue=
- .Form=
- .Printer=
- .Stay_home=

.Customer_Name>

This directive specifying the customer's name to CHART/II. Must be entered exactly as specified on your licensing agreement. Must be specified first and immediately preceding the .CUSTOMER_LOCATION> command.

.Customer_Location>

This directive specifies the customer's location to CHART/II. Must be entered exactly as specified on your licensing agreement, enclosed in quotes ("). Must be specified immediately following the .CUSTOMER_NAME> command and immediately preceding the .CUSTOMER_SW_LICENSE> command.

.Customer_SW_License>

This directive specifying the customer's software license number to CHART/II. Must be entered exactly as specified on your licensing agreement. Must be specified immediately following the .CUSTOMER_LOCATION> command and immediately preceding the .CUSTOMER_USE_LIMIT> command.

.Customer_Use_Limit>

This directive specifying the customer's user limitation code to CHART/II. Must be entered exactly as specified on your licensing agreement. Must be specified immediately following the .CUSTOMER_SW_LICENSE> command and immediately preceding the .CUSTOMER_CONFIG_CODE> command.

.Customer_Config_Code>

The directive specifying the customer's configuration code to CHART/II. Must be entered exactly as specified on your licensing agreement. Must be specified immediately following the .CUSTOMER_USE_LIMIT> command and immediately preceding the .CUSTOMER_CONFIG_ID> command.

.Customer_Config_ID>

This directive specifying the customer's configuration identifier to CHART/II. Must be entered exactly as specified on your licensing agreement. Must be specified immediately following the .CUSTOMER_CONFIG_CODE> command.

.Data_Base

The .DATA_BASE xx = `full_data_base_name` command is used here to define abbreviated database names in the display files themselves. The abbreviation xx can be used in the display file instead of the name of the database itself. The string `full_data_base_name` may be any database name.

.High_Pri=

The high priority 'n' is used to read new display files from disk. Recommended values are between 4 and 10. Values above 15 are not recommended.

.Low_Pri=

The low priority 'm' is the normal running priority of CHART/II at display-update time. This number should normally be 4.

.Password=

The .PASSWORD= command (located in CTPINI.CTP) is used to specify a system password of up to six characters. If a blank password is specified, the display can be unlocked with the `U` command. If `XYZ` is the password, the display can be unlocked with UXYZ. When a display is locked, all fields with the /L switch are locked from VALUE mode changes and the VALUE mode cursor will not stop on the field. The following is a command example.

```
.PASSWORD = CHTSYS
```

.Queue=

The `.QUEUE=queue` command establishes the print queue for hardcopy output from the Chart terminal. The argument is the name of a valid print queue on the machine running CHART/II. If this command is missing, the default is SYS$PRINT.

.Form=

The `.FORM=form` command sets the form name in the print request when hardcopy output from a chart terminal is produced. If this command is missing, the default form for the print queue is used. Refer to the DCL commands DEFINE /FORM and PRINT /FORM for more information.

.Printer=

The `.PRINTER=type` command sets the printer type for producing hardcopy output with the proper internal format. The valid printer types are as follows.

- LA50
- LA75
- LA100 (the default type)
- LA210
- LA324
- LJ250
- LN03

.Stay_home=

This command causes the display file that was used to start CHART/II to become locked in as the 'Home' display. If this command is missing, the 'Home' display changes to the current display upon use of the keylink to change displays or if a numbered display is requested.

General

You can select a different plotting line color or character depending on the value of the variable being plotted. You could, for example, use dots for value below 80% and diamonds for values over 80%.

Since the VT241 limits each screen to a total of 4 colors it is probably better to use different plotting characters for each line than to use different colors. This also works better for black and white hardcopy.

The VT340 improves the plotting capabilities tremendously. The resolution on the screen, the 16 colors and the improved printer output are the main enhancements.

Graphic Display Creation

All Chart displays are created with TPU or any other editor capable of generating ASCII text files. All of the text mode display creation features are supported with the following additions.

The origin of each grid is specified with the grid origin location as follows.

{Gn}

Where n is a number from 0 to 19 indicating the grid number that is defined along with the necessary scaling information in the command section of the display. You can have up to twenty {Gn} origins on each display. The origin of each grid is defined by the { character and the grid size information is specified following the display section of the file (lines 1 thru 22) by grid size and control definition.

Line Versus Point Plotting

If you are plotting X,Y pairs as a scatter plot, the best technique is to use point characters. If you choose to use line plotting, you should place the independent variable on the horizontal X axis and the dependent variable on the vertical Y axis.

Line plots of curves that have more than one Y value for a given X value are not true functions and the REPAINT HISTORY will not necessarily redraw the original data properly. In this case, it is better to use point characters than lines.

Even if you are plotting lines that have only one Y for each X value, the X value should still always change in the same direction or the lines connecting the samples will reverse. Again, line plots are best used in cases where the X variable covers a particular range (say 50 to 100 degrees) and the Y variable changes depending on X (the temperature).

Clearing Grids

The -100 pen control code is reserved for grid clearing. When your application program determines that the grid should be erased, it should place a -100 in the variable designated as the pen control variable. When Chart has completed the clearing of the specified grid, it returns the -100 code in the CHART_DONE variable. This is the indication to your program that the grid is ready for more data.

Clearing Grids (cont)

If you are plotting more than one X,Y pair on the same grid, you should set all other pen control variables to -1 before clearing the grid. Only one of the pen control variables has to be set to -100 to clear the grid in this case.

Erasure Bar

If you enable the erasure bar for an X,Y pair, you may be able to forget about clearing the entire grid. The erasure bar automatically clears a portion of the grid immediately in front of the current X,Y line being drawn. The size of the erased section is a minimum of one vertical grid section as specified by the number of vertical lines in the grid.

The erasure bar works well when plotting lines and when using the auto pen lift. Point characters can also be used.

Once the erasure bar is established, it will continue in front of the current X value. If X changes direction, the erasure bar will attempt to recover after two consecutive X moves in the new direction. Repeated X reversals may cause seemingly erratic erasure.

If there are no vertical grid lines, the erasure bar will not function. If there are two vertical grid lines, one at the origin and one at full scale, the entire grid will be erased when the first point is plotted or when X reverses direction.

Creating a Screen Hard Copy

A hard copy of the current CHART/II screen image can be generated by two(2) methods. The user may push the keypad <6> key (when the keyboard is unlocked), or the .PRINT_SCREEN directive can be used in the display file to link a CRISP/32 database variable that can 'trip' a screen copy based on logic conditions. When this occurs, a pixel image disk file is created from the VT240/241/330/340, and can be stored or printed. The ASCII file is in SIXEL-1 format compatible with the LA50 and LA75 bit map protocol.

When activated, the background screen color of the DEC terminal will change to black. A header will appear at the top of the screen giving the name of the display, name of the image file, and the time the image file was made. The screen will stay this way for a time period of 2 to 5 minutes. The creation of the image file is complete when the user's display resumes updating. The VT300 series terminals will return to the original color, also.

For more information on the .PRINT_SCREEN directive, see the chapter on display creation.

History Repaint

There are several ways to use the history repaint feature. The two main categories are as follows.

1. The user can load the history buffer with data and specify the buffer length in the length variable. (Refer to the .XY command.) In this mode, you should set the DATA_CONTROL LSB to zero so that CHT does not try to place data in the buffer.

   With this feature, the user has full control of the data being drawn by the repaint. Evenly spaced Y values are loaded into the buffer and the X values are implied by the buffer length. All Y values in the buffer are placed on the grid, evenly spaced in the X direction. Any values of -9999 located in the buffer are ignored. All other data are plotted using the current grid scaling.

2. The second method of using the history buffer is to allow CHT to store the current plotting data in the buffer as it occurs. The Y values are placed in the buffer as though the buffer were evenly spread out across the grid. This means that the ultimate resolution of the history repaint is determined by the size of the buffer. Any missing Y values are indicated in the buffer as -9999.

The two points to remember are: the DATA_CONTROL variable LSB determines whether CHT or the user places data in the buffer; and, the DATA_LENGTH variable determines the size of the buffer, if any. A zero for DATA_LENGTH indicates that there is no buffer and that a history repaint cannot be performed.

Lifting the Pen

If you are plotting lines across a grid and your X (independent) variable reaches the end of the grid, you will normally want to lift the pen (pen code -1) or disable the plotting (pen code 0). Depending on the purpose of the graph, you may want to reset the independent variable to the origin, put the pen down (pen codes -2,-3,-4) and continue drawing. If you reset the independent variable to the origin with the pen down, you will produce a line from the current point back to the origin. This is obviously not a consideration if you are using point characters for plotting.

The automatic pen lift feature can be used to force the pen up when the X variable changes direction. If you choose this feature (see DATA_CONTROL_INDEX above), you don't have to change the pen control to the pen-up code. Simply reset the X coordinate to the beginning of your graph and Chart will lift the pen. The auto pen lift should not be used with point plots or graphs where there can be more than one Y for a given X.

Notes:

Sample Display File

```
CHART ON {8} SAMPLE RATE= {7} TOTAL PASSES FOR THIS SAMPLE=

{6}
AVERAGE LOGIC CYCLE TIME= {9}
{4} 10
9
PASSES 8
PER 7
SECOND 6
5
4
3
2
1 {G0}
{5} 0
0 TIME IN SECONDS SINCE LAST GRID CLEAR 100


X,Y={1},{2} PEN= {3}




.DATA_BASE=CHART,CHART1
! 10 CHRS HI, 100 WIDE, 11 XY GRID LINES
.G0= 10,100,11,11,999
.XY= 0,1,2,3,11,12,13,14,0,0,1,50
! GRID=0,X,Y,PEN, Origin=(0,0),
! Engr.  units per character in X and Y.
.SYMBOL
1/1:T1.X
2/1:T1.Y
3/1:T1.USER.PEN/K=C:-100
4/1:T1.TOP
5 /1:T1.BOT
6 /1:.PASSES
7 /1:T1.TMR
8 /1:T1.CHART.ON
9 /1:CYCLETIME
11 /1:DONE
12 /1:CONTROL
13 /1:BUFFER
14 /1:BUFFER.LENGTH
999/1:GRID.CONTROL
```

Notes:

Appendix B

Escape Sequences

This appendix summarizes some of the escape sequences which may be used to control various features of the DEC VT-series terminals. Please consult the appropriate programming manual for the terminal(s) you are using to determine the support of these escape sequences by your terminal.

In the examples below, the string <ESC> represents the ASCII escape character. An escape may be inserted into your TPU or EDT editing buffer by typing GOLD (PF1), 27 (using the keys above the alpha keys), GOLD, and SPECINS (3 on the numeric keypad). If you are using the TPU editor, an escape may also be inserted by typing Ctrl/V followed by Ctrl/[.

Terminal Modes

<ESC>[?3l	Set terminal to 80-column mode
<ESC>[?3h	Set terminal to 132-column mode

Cursor Positioning

<ESC>[3;1H	Move cursor to line 3, column 1

Character Attributes

<ESC>[1m	Display following text in bold
<ESC>[4m	Display following text underlined
<ESC>[5m	Display following text blinking
<ESC>[7m	Display following text in inverse video
<ESC>[m	Display following text in normal video

Attributes may be combined as shown below

<ESC>[1;5m	Display following text in bold & blinking

Line Attributes

<ESC>#3	Double-width, double-height line (top)
<ESC>#4	Double-width, double-height line (bottom)
<ESC>#6	Double-width, single-height line

Color Selection (VT241, VT340 only)

On graphics terminals with color monitors, the text and background colors may be changed by using a series of ReGIS graphics commands. To do this, use a line similar to the following at the beginning of your display file(s). See the sample file [CRISP.CH2.DEMO]DIRECTORY.DSP for an example of this color setup. The example below has been broken to fit on the page, however, it should be a single line in an actual display file.

```
.<ESC>[?3h<ESC>PpS(C0,M0(AH0L50S100)M1(AH120L46S72)
 M2(AH180L65S60)M3(AH0L90S0),I0,E)W(I3)<ESC>\<ESC>[3;1H
```

The characters Mn(...) shown above specify a which color attribute should be changed. The terminal color attribute usage is shown below.

Mode	**VT241**	**VT340**
Normal	M3 on M0	M7 on M0
Reverse	M0 on M3	M0 on M7
Bold	M2 on M0	M15 on M0
Blink	M2 on M0 / M1 on M3	M15 on M0 / M8 on M7
Bold & Blink	M0 on M2 / M3 on M1	M0 on M15 / M7 on M8

Color Selection (cont)

Text-mode colors on the VT340 terminal can also be changed using the terminal's built-in setup facility.

The characters AHnnnLnnnSnnn shown previously specify a color using the Hue, Lightness, Saturation (HLS) system. The following are specific colors.

Color	**H**	**L**	**S**
Black	0	0	0
White	0	0	0
Red	120	46	72
Green	240	50	100
Blue	0	50	100
Cyan	300	50	60
Magenta	160	50	100
Yellow	180	65	60

Special Graphics Characters

The VTxxx-series terminals are capable of displaying some special graphics characters such as line drawing characters by exchanging the lower-case characters for the special graphics characters. Some of the displays in [CRISP.CH2.DEMO] use this feature to draw a box around portions of the display. The characters used to control this capability are shown below.

<ESC>)0 Load the special graphics character set. This sequence should be used once at the beginning of the display file.

Ctrl/N Map special graphics characters into lowercase characters. In TPU or EDT, insert Ctrl/N with <PF1>14<PF1><KP3>.

Ctrl/O Restore the normal lowercase characters. In TPU or EDT, insert Ctrl/O with <PF1>15<PF1><KP3>.

After a Ctrl/N, you may use the following lowercase characters to specify their corresponding special graphics characters.

f	○	Degree sign
g	±	Plus/minus sign
j	┘	Lower right corner
k	┐	Upper right corner
l	┌	Upper left corner
m	└	Lower left corner
q	▬	Horizontal connector
t	├	Left side tee
u	┤	Right side tee
v	┴	Bottom side tee
w	┬	Top side tee
x	▌	Vertical connector

Terminal Setup Features

Any terminal used for CHART/II should be setup for the attributes below by using the SET commands. You can use the SHOW TERM command to list the attributes of your terminal.

- NOWRAP
- LOWER
- ECHO

A VT102 SETUP B should look like this:

```
1100 0011 0000 1101 1010 1000 0010
```

A VT100 must have AVO and SETUP B should look like this:

```
0100 0011 0000 0010
```

A VT240/1 or VT340 terminal should be setup with the following features:

- NO AUTO WRAP
- NO TEXT CURSOR
- NO MARGIN BELL

Notes:

Data Type Summary

PLOTTING	ARGUMENT	VAL ALLOW.	
ARGUMENT	DESCRIPTION	VAR	TYPES
--------	-----------	---	---------
	For the 'G=' plotting command:		
H_CHAR%	Height of Grid	VAL	
W_CHAR%	Width of Grid	VAL	
H_LINE%	Number of horizontal Lines	VAL	
W_LINE%	Number of vertical Lines	VAL	
GRID_CONTROL%	Grid control variable	VAR	Word
	For the 'XY=' plotting command:		
G	Grid number	VAL	
X	The X variable	VAR	Word
	Float		
			Long
Y	The Y variable	VAR	Word
	Float		
			Long
P	Pen control variable	VAR	Word
DONE	Plot done variable	VAR	Word
DATA_CON	Historical buffer control	VAR	Word
DATA_START	Historical buffer start	VAR	Word
			Float
			Long
DATA_LEN	Length of History Buffer	VAR	Word
X0	The X zero value	VAL	
Y0	The Y zero value	VAL	
X/CHAR	The X scale per/char	VAL	
Y/CHAR	The Y scale per/char	VAL	

The purpose of this table is to summarize the arguments used in the plotting commands described previously. The word VAL indicates that the argument is a value and not an index to a variable. The word VAR indicates that the argument is an index to a variable in the database.

The data type column specifies the type of database variable that can be used for the argument.

Notes: