DOT-TSC-OST-71-1

GUIDE TO THE USE OF THE DYNAMIC DISPLAY SOFTWARE SYSTEM

January 1971

USER'S MANUAL

Prepared for

OFFICE OF THE SECRETARY

TRANSPORTATION SYSTEMS CENTER
55 Broadway
Cambridge, Mass. 02142

DOT-TSC-OST-71-1

GUIDE TO THE USE OF THE DYNAMIC DISPLAY SOFTWARE SYSTEM

January 1971

Transportation Systems Center
Department of Transportation
Cambridge, Mass. 02142

TABLE OF CONTENTS

1.0 INTRODUCTION

The Dynamic Display Software System (DISS)[1] was designed and implemented to provide most scientifically oriented personnel with the ability, via a highly interactive and interpretive language, to create desired displays and specify dynamic parameters in order to simulate a particular environment. The command language and data structure necessary to accomplish this goal are embodied in DISS.

This manual is intended to provide the user with an easy reference to the commands and use of DISS, and to act as a guide when designing and entering a display into the display system via DISS. This document is arranged with a summary of commands, their format, and their uses, as well as an alphabetical and functional quick reference list in the appendix.

The user need not be a highly sophisticated programmer to create desired displays successfully. However, the aid of a more advanced programmer may be required to explain the interface of the display system to dynamic processes, e.g., a simulation. Once learned, however, the procedure is straightforward and could be utilized by any technically oriented person.

DISS was designed with N. A. S. A. funds over the years 1968-1970 by the Wolf Research and Development Corporation. Special thanks for this work go to Mr. Albert V. Shortell, who worked harder than could have been expected on the project, and to Mr. John M. Hetherington, now deceased, who added his verve and effort to many of the programs in the system. Other personnel involved were:

WOLF RESEARCH & DEVELOPMENT CORP. -AND-AZREX, INC.

Mr. Hiram French

Mr. William Albertson

Mr. Robert Johnson

Mr. Jon Meads

Mr. Frank Graboski

Mr. Algert Jonuskis

Mr. Phillip Sweet

Mr. Donald Gray

Mrs. Linda Soderman

NASA/ELECTRONICS RESEARCH CENTER - AND - DOT/
TRANSPORTATION SYSTEMS CENTER

Dr. Daniel Kennedy

Mr. Richard Wright

Richard D. Wright

1970 Cambridge, Mass.

2.0 SYSTEM OPERATION

The Dynamic Display Software System $(DISS)^2$ must be used in conjunction with the Disk Operating System (DOS) . DISS uses several subroutines internal to DOS that are available for general use e.g., SAVE and RESTOR. The SAVE subroutine is used in the input/output phase, and the RESTOR subroutine is used in the simulation and input/output Phase.

2.1 System Initialization

The user must be "attached" to a User-File Directory (UFD) that contains the following files:

a) DISSYS or equivalent - core image of create/edit phase.

b) OUTPUT - core image of mini-compiler and output routine.

c) CREATE - portion of DISSYS overlaid by OUTPUT.

d) SIMLTE - core image of simulation phase.

All data structures will be saved or reloaded using the attached Open-File Directory.

The procedure to initiate DISS is:

a) Load in DOS object tape using the paper tape reader.

b) System Reset and manually start at 30000_8.

DOS will then type out "OK" and the proper sequence will be:

```
USER-STARTUP
DOS-OK
USER-ATTACH  UFD 1        File directory name
DOS-OK
USER-R DISSYS             Restore create/edit phase
DOS-GO
DDSS-:                    Ready for user language commands
```

Omit step "a" and the STARTUP command if DOS had previously been loaded.

2.2 Monitor Operation

The Monitor is the main program for the Simulation Phase. Its functions include:

a) The input of data files from disk - The user types in the data file name he desires to simulate, and Monitor calls the DOS RESTOR subroutine to read in the data structure.

b) The processing of interrupts - The Monitor services interrupts from both the photo pen and real time clock. Clock interrupts initiate the update cycle, while photo pen interrupts call the user supplied photo pen interrupt routine.

c) Device polling - The Pilot Control Station, Display Generator, and PDP-10 do not create interrupts. Instead, they are polled (under user option) during each update cycle.

Provision is made in the Monitor for linking the above routines. However, the user should consult with system programmers at the display laboratory to determine which routines are currently implemented.

The user enters the simulation phase by either the "O" option in the SAVE and RELOAD command (See Section 3.6) in the create/edit phase, or by typing RESUME SIMLTE using DOS.

An example of the dialogue once Monitor has been entered follows:

DATA = XAMPLE, DT = 100↵

POLL DG = Y, PPI = N, PDP = N, PCS = Y

Underlined characters indicate Monitor response. First, the user types in the data file name, which must be in the attached User File Directory. Next, the update cycle time is typed in milliseconds followed by a carriage return. The Monitor then asks if the Display Generator, Photo Pen Interrups, PDP-10 computer, and Pilot Control Station are to be polled. User response will be either Y (yes) or N (no).

If the update process takes longer to complete than the update cycle time, Monitor types an appropriate error message indicating where the time out occurred, e. g. , TIME OUT IN MODEL. The simulation phase can then be reentered by depressing the computer start button and typing in parameters as shown above but with a longer cycle time.

2. 3 Application Program Attachment

A register bank provides the interface between the user application program and the Simulation Phase. Math models, Background and I/O routines make up the user application program. For some simulations, any or all of these routines may be eliminated. This is possible because of the mini-compiler which produces executable code for dynamic expressions entered during the create/edit phase and updated during the simulation phase. All references to the dedicated register bank are via a labelled common statement. In DAP-16 assembly language, the reference would be:

Internal array name COMM IREG

While in FORTRAN the reference would be:

COMMON/IREG/Internal array name, Defn

The procedure to attach application programs is as follows:

1. Generate program object tape.
2. Load the following tapes using the Linking Loader.
   a) Monitor (required)
   b) Update (required)
   c) Background (optional)
   d) Model (optional)
3. Save this group of programs using DOS.
4. Run this file with the data structure in place of SIMLTE.

The load must not exceed $11,500_8$ as high location; otherwise it will be partially overlaid by the data structure.

3.0 USER LANGUAGE

The user language enables the creation of unique graphical representations of information via the ASR-35 keyboard. Components are displayed as they are created, providing positive feedback. The system differs from a simple computer aided drafting system in that graphical entities can be defined algebraically in terms of system simulation variables. Alterations of these variables by the simulation program are reflected by changes in the information displayed during the simulation phase.

The user language commands are organized in a hierarchical order. The highest class of items defined are frames. A frame consists of all graphical and associated dynamic information to be displayed at a particular time. Frames are mutually exclusive, and no two frames may appear at once. All registers are defined at this level. Indicators are used to specify certain display parameters such as blank/unblank, display device to use, and skip on condition. This parameter then applies to all elements of the indicator. Each indicator is made up of a series of entities which define the geometrical figures seen on the CRT. Conditions much like those for the indicator may be defined for the entity. In addition, an entity may be defined as static or as dynamic, so that the entity will rotate and move its position on the display screen. Each entity is controlled by the conditions defined for the indicator of which it is a part as well as by conditions defined specifically for that entity. However, in the event of a conflict, conditioning code entered at the entity level takes precedence.

When issuing commands, the user enters the command mnemonic following the colon (:) printed by the system. A space must then be typed, followed by the argument(s) required and a carriage return. The arguments are in most cases separated by commas or spaces. Following a dynamic expression, a semi-colon is used.

EXAMPLE:

```
:DR expr;N,S↵
```

Comments may be inserted by the user by issuing as asterisk (*) immediately after the colon. The asterisk is then followed by a space and the comment text. Rubout will delete the last character typed and control "X" will delete the entire line.

There are several types of user language commands, some of which are used on several hierarchical levels (e.g., the skip commands) and others which are used on one level only (e.g., most of the drawing commands). See Appendix C for an example of the use of the system language in the creation of display information.

3.1 Data Structuring Commands

The three data structuring commands, FRAME, INDICATOR, and ENTITY define successively lower working levels of the language. The levels must be specified in hierarchical order: first frame, then indicator, then entity. In the following commands, ID is an identification number from 1 to 64.

3.1.1 FRAME

:F ID

The identified frame is retrieved or created. In the case of a retrieved frame, the entire frame, including indicators, entities, components, and conditioning blocks, is retrieved and displayed; and the current frame is detached. If the identified frame has not been previously assigned, an empty frame block with the given ID is created; and nothing is displayed.

3.1.2 INDICATOR

:I ID

The identified indicator is retrieved or created. If previously assigned, the indicator is opened for input. If the indicator has not been previously assigned, a new indicator block with the given ID is created and attached to the current frame block.

3.1.3 ENTITY

:E ID

The identified entity is retrieved or created. If previously defined, both the conditioning and component (display) blocks of the entity are opened for input. For a new entity, an entity block with the given ID is created and attached to the indicator.

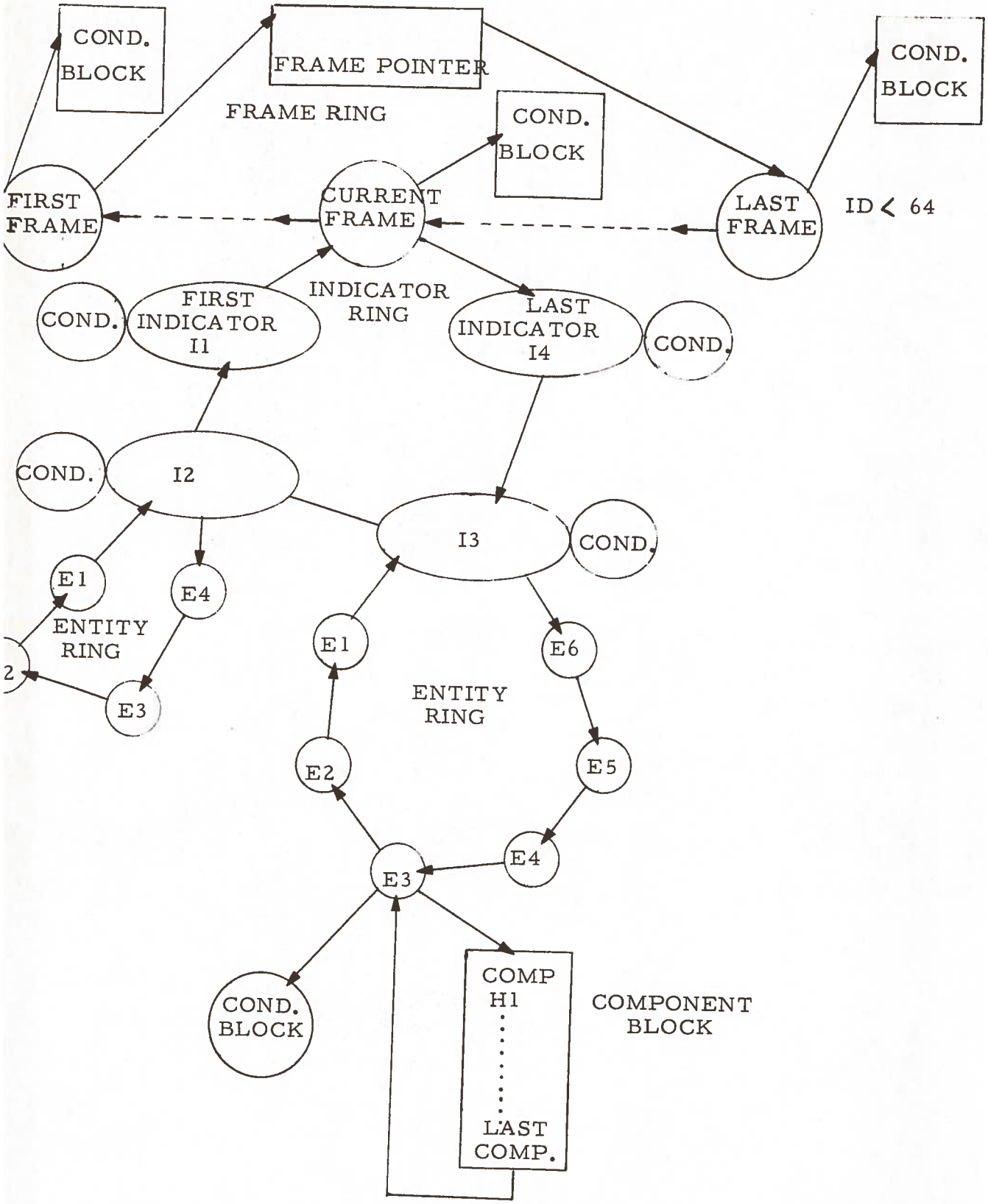


Figure III. 1
Ring Organization of the Data Structure

3.2 Register Definition

DEFINE REGISTER

:DRE RXX, expr (where XX = number from 0 to 99).

A frame level command, define register causes a register RXX to be defined by the expression, expr. Registers are defined after frame specification but before indicator specification. They are used as variables in expressions. Registers 0 through 9 contain model parameters and pilot control status. (See Section 2.3.) Expr may be any Fortran compatible arithmetic expression containing variables and/or integer constants, but the variables must be register names.

In the simulation phase during display update, the expressions defining the user registers are evaluated and updated. Before evaluation, each register contains the value assigned to it on the last pass. The registers are initially zero.

3.3 Editing Commands

3.3.1 DELETE FRAME

:D F ID

3.3.2 DELETE INDICATOR

:D F ID, I ID

3.3.3 DELETE ENTITY

:D F ID, I ID, E ID

3.3.4 DELETE CURRENT ENTITY

:D

The highest level block to be deleted and all lower levels are removed from the display list and returned to free storage. After execution of a delete command, the operating level becomes one higher than the highest level at which deletion occured. For example, after a delete indicator command, the system returns to the frame operating levelfor the frame to which the deleted indicator was attached.

3.4 Conditioning Commands

Conditioning commands for indicators and entities are entered into conditioning blocks for each of these levels. During the simulation phase, these conditioning blocks are used by the user language processor to update the display list. Conditioning commands, therefore, have no effect upon the create-edit phase display list; and so produce no changes in the displayed items until entry into the simulation phase.

It should be noted that conditioning commands are executed in the order encountered. This means that the commands will be executed as follows:

First Frame Level Conditioning Command,

.

.

.

Last Frame Level Conditioning Command,

.

.

.

First Conditioning Command of Last Indicator,

.

.

.

Last Conditioning Command of Last Indicator,
First Conditioning Command of Last Entity of Indicator,

.

.

.

Last Conditioning Command of Last Entity of Indicator,

.

.

.

Last Conditioning Command of First Indicator,
First Conditioning Command of Last Entity of Indicator,

.

.

.

Last Conditioning Command of First Entity of Indicator,
and repeat at next update.

The following conditioning commands may be given on either the indicator or entity levels. Commands given on the indicator level affect all entities connected to that indicator.

3.4.1 INTENSITY

:IN I

The intensity command is inserted in the conditioning block. At simulation time, the intensity of the appropriate indicator or entity is set to the level specified by I (1 to 7, where 1 is the lowest intensity and 7 the highest).

3.4.2 TEXTURE

:TX T

Where T = Texture:

S	Solid
D	Dot
DA	Dash
DD	Dot-Dash

The texture command for indicator or entity is inserted in the conditioning block. At simulation time, the texture of the indicator or entity is set to that specified. If texture is not specified, a solid image is displayed.

3.4.3 BLINK

:B

The blink command for indicator or entity is inserted in the conditioning block. At simulation time blinking of the indicator or entity components begins.

3.4.4 UNBLINK

:U

The unblink command for indicator or entity is inserted in the conditioning block. At simulation time, if that indicator or entity is blinking, the blinking is stopped.

3.4.5 OFF

:OF

The command to disconnect the indicator or entity from the display list is inserted in the conditioning block. At simulation time the appropriate indicator or entity has the blank bit set in its corresponding z-axis control word for the ADDS/900. Update of the blanked item continues.

3.4.6 ON

:ON

The command to unblank the indicator or entity in the display list is inserted in the conditioning block. At simulation time the appropriate indicator or entity has the blank bit reset in its z-axis control word.

3.4.7 NEW FRAME

:N ID

The command to stop display of the current frame and start display of the specified frame is inserted into the conditioning block. When the command is encountered at simulation time, the display of the current frame is discontinued, replaced by that of the appropriate frame. Update of a non-displayed frame does not take place.

3.4.8 TRANSLATE

3.4.8.1 When given in the indicator mode:

:T Δx, Δy

Where Δx = x-component of translation

Δy = y-component of translation

When this command is encountered at simulation time, all entities in the current indicator are translated by the specified Δx, Δy. The current set points are reset.

3.4.8.2 When given in the entity mode:

:T x-expr; y-expr

Where x-expr = dynamic expression for the x-component of translation.

y-expr = dynamic expression for the y-component of translation.

When an entity level translate command is encountered during the simulation phase, the x and y expressions are evaluated; and the entity is translated by the calculated amount.

The following conditioning command may be given on the entity level only:

3.4.9 ROTATE

:R x-expr; y-expr; θ-expr

Where x-expr = dynamic expression for the x-coordinate of the center point relative to the current beam position.

y-expr = dynamic expression for the y-coordinate of the center point relative to the current beam position.

θ-expr = dynamic expression for the angle of arc.

During the simulation phase the entity is rotated about the center point specified by x-expr and y-expr relative to the current beam position. The angle of rotation is specified by θ-expr.

SKIP Commands

There are six SKIP commands which the user may intersperse in the conditioning block command tests to control the sequencing of events in the simulation phase. These commands can be given on either the indicator or entity level. A and B represent predefined registers, and N is the number of conditioning block statements to be skipped (1-127 octal). See Table 1 for the number of conditioning block commands generated by the user commands so that the correct N value can be found.

3.4.10 SKIP

:SK N

N conditioning block statements are unconditionally skipped.

3.4.11 SKIP IF EQUAL

:EQ A, B, N

If register A = register B, N conditioning block statements are skipped.

3.4.12 SKIP IF GREATER THAN

:G A, B, N

If register A > register B, N conditioning block statements are skipped.

3.4.13 SKIP IF GREATER THAN OR EQUAL

:GE A, B, N

If register A ≥ register B, N conditioning block statements are skipped.

3.4.14 SKIP IF LESS THAN

:LT A, B, N

If register A < register B, N conditioning block statements are skipped.

3.4.15 SKIP IF LESS THAN OR EQUAL

:LE A, B, N

If register A ≤ register B, N conditioning block statements are skipped.

TABLE 1 Conditioning Block Commands Generated by User Commands

	COMMAND	NUMBER OF COMMANDS
	B	1
*	BL	1 or 2
	DR	1
	EQ	2
	G	2
	GE	2
	IN	1
*	L	1 or 2
	LE	2
	LT	2
	N	1
	OF	1
	ON	1
*	P	1 or 2
	R	1
	SK	1
	T	1
	TX	1
	U	1

OTE:

* Number of Conditioning Block Commands Generated = Number of Expressions in Command.

3.5 Drawing Commands

The drawing commands are used to create and position images on the face of the CRT. Each command causes display information to be placed in the component block of the current entity, causing the information to be displayed immediately.

In several drawing commands, Δx, and/or Δy can be specified dynamically. These commands are Point, Blank Line, and Line.

When specifying a component dynamically, the form

N #expr;

is used. N is the static value of the component displayed during the create phase. During the simulation phase, the expression expr is evaluated; and the component is displayed using this value. Expr may be any Fortran expression consisting of variables and integer constants only. The allowable variables are names of the predefined registers, R 0 through R 99. The expression is checked for errors between the create and simulation phases.

All dynamic expressions must be terminated by either a semi-colon or a carriage return. If no dynamic expression exists for a component, the #expr may be omitted; and static value of the component will be used.

3.5.1 SET INITIAL TEXTURE, INTENSITY

:STI T, I (or I, T)

Where T = texture:

S	solid
D	dot
DA	dash
DD	dot-dash

I = intensity (0 - 7) where 7 is the highest intensity.

The information displayed during the create phase and initially at simulation time is displayed with the specified texture and intensity. If the working level when the command is given is the indicator mode, all entities of that indicator

are set to the given texture and intensity. If given at the entity level, the command affects only that entity.

3.5.2 SET POINT

:S X, Y where X = absolute x coordinate
Y = absolute y coordinate

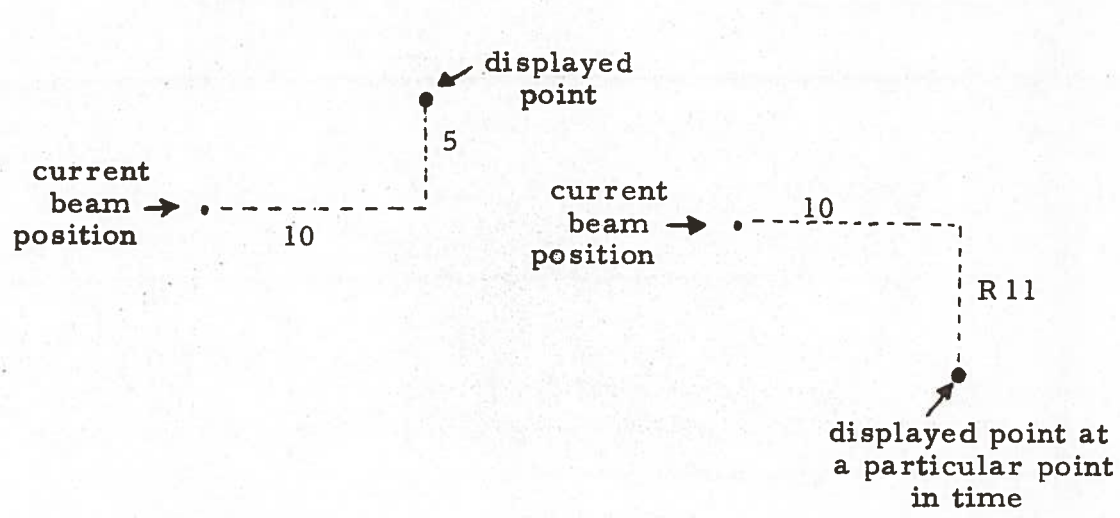
The beam is positioned at the point (x, y) for that entity. (The center of the CRT is the point (0, 0).) Only one set point command is effective for each entity, i.e., succeeding set point commands overwrite the previous value, causing the entity to be shifted to newest values.

3.5.3 POINT

:P Δx #x-expr; Δy #y-expr

During the create phase a point is displayed at the point (Δx, Δy) relative to the current beam position. If the point is specified dynamically, the expressions are saved for future processing during the simulation phase. The point is then displayed at simulation time at the evaluated position.

Example: If the command :P 10, 5#R11 is given, the following will result:



3.5.4 BLANK LINE

:BL Δx #x-expr; Δy #y-expr

Initially a blank line is drawn between the current beam position and the point (Δx, Δy) relative to the current beam position. If one or two dynamic expressions are included in the command, they are saved for future processing during the simulation phase when a blank line will be drawn to the evaluated point.

3.5.5 LINE

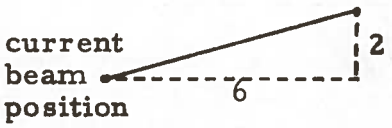
:L Δx #x-expr; Δy #y-expr

During the create phase a line is displayed connecting the current beam position with the point (Δx, Δy) relative to the current beam position. If one or two dynamic expressions are included in the command, they are saved for future processing during the simulation phase when a line will be drawn to the evaluated point,

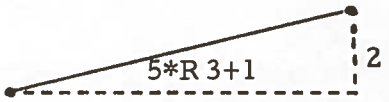
Example:

:L 6#5*R3+1;2

Create Phase



Simulation Phase



If R3 changes with time, the line will move each time the expression is updated.

3.5.6 ARC

:A Δx, Δy, θ

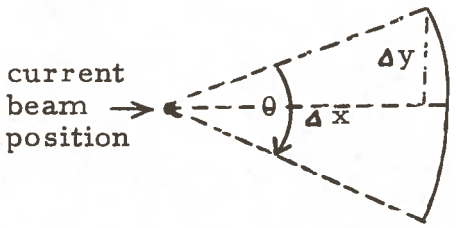
Where Δx = x-coordinate of radius relative to current beam position; or, if Δy and θ are not given, Δx is length of the radius for the full circle.

Δy = y-coordinate of radius relative to current beam position.

θ = angle of arc in degrees.
If θ is not given, it is assumed to be 360^{o}.

An arc is drawn using the current beam position as a center point, a radius of length $\sqrt{(\Delta x)^2+(\Delta y)^2}$, and θ as the angle of arc. After the arc has been drawn, the beam position is returned to the initial point.

Example: :A Δx, Δy, θ



3.5.7 TICK MARKS FOR LINE

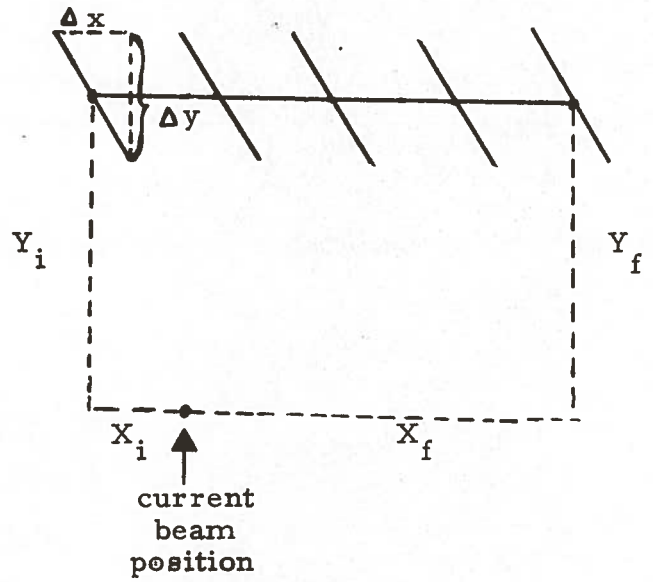
:TM N, X_i, Y_i, X_f, Y_f, Δx, Δy

Where
N = number of tick marks
X_i = x-coordinate of initial tick mark location relative to current beam position.
Y_i = y-coordinate of initial tick mark location relative to current beam position.
X_f = x-coordinate of final tick mark location relative to current beam position.
Y_f = y-coordinate of final tick mark location relative to current beam position.
Δx = x-coordinate of tick mark (length)
Δy = y-coordinate of tick mark (length)

Tick marks are displayed along the line given by the points (X_i, Y_i) and (X_f, Y_f) relative to the current beam position.

Tick marks are drawn such that the given line bisects them. If no Δ y is given, Δ x is assumed to be the length of the tick mark; and the tick marks are drawn perpendicular to the given line. If neither Δ x nor Δ y is given, tick marks with a length of 12 raster units are drawn perpendicular to the given line. A line need not be displayed between (X_i, Y_i) and (X_f, Y_f) in order for tick marks to be drawn. Therefore, the tick mark command is useful in drawing columns or grids on the CRT. Example:

:TM 5, X_i, Y_i, X_f, Y_f, Δx, Δy



3.5.8 TICK MARKS FOR ARC

:TMA Δx, Δy, N, L, θ

Where Δx=x-component of initial radial vector relative to current beam position.

Δy=y-coordinate of initial radial vector relative to current beam position.

N = number of tick marks

L = length of tick marks

θ = angle of arc (in degrees) which is to be divided by tick marks

N tick marks of length L are displayed for the arc with current beam position as center point. If the angle of arc is not given, it is assumed to be 360^{o}. If neither angle of arc nor length of tick marks is given, then tick marks 12 raster units long are drawn around a full circle. All tick marks are drawn perpendicular to and bisected by the arc. The final beam position is at the center point. The arc itself need not be displayed for tick marks to be drawn.

3.5.9 SCALES FOR LINES

:SL X_i, Y_i, X_f, Y_f, A, B, N, S

Where X_i = x-coordinate of initial character relative to the current beam position.

Y_i = y-coordinate of initial character relative to the current beam position.

X_f = x-coordinate of final character relative to the current beam position.

Y_f = y-coordinate of final character relative to the current beam position.

A = initial label value

B = final label value

N = number of labels

S = character size (1 - 4), where 4 is the largest.

N Numeric labels with values from A to B are displayed as scales on a line. The final beam position is (X_f, Y_f) relative to the initial beam position. A line itself need not exist for scales to be displayed.

3.5.10 SCALES FOR ARCS

:SA Δx, Δy, A, B, N, S, D, θ

Where Δx = x-coordinate of initial radial vector relative to current beam position.

Δy = y-coordinate of initial radial vector relative to current beam position.

A = initial label value

B = final label value

N = number of labels

S = size of characters (1 - 4)

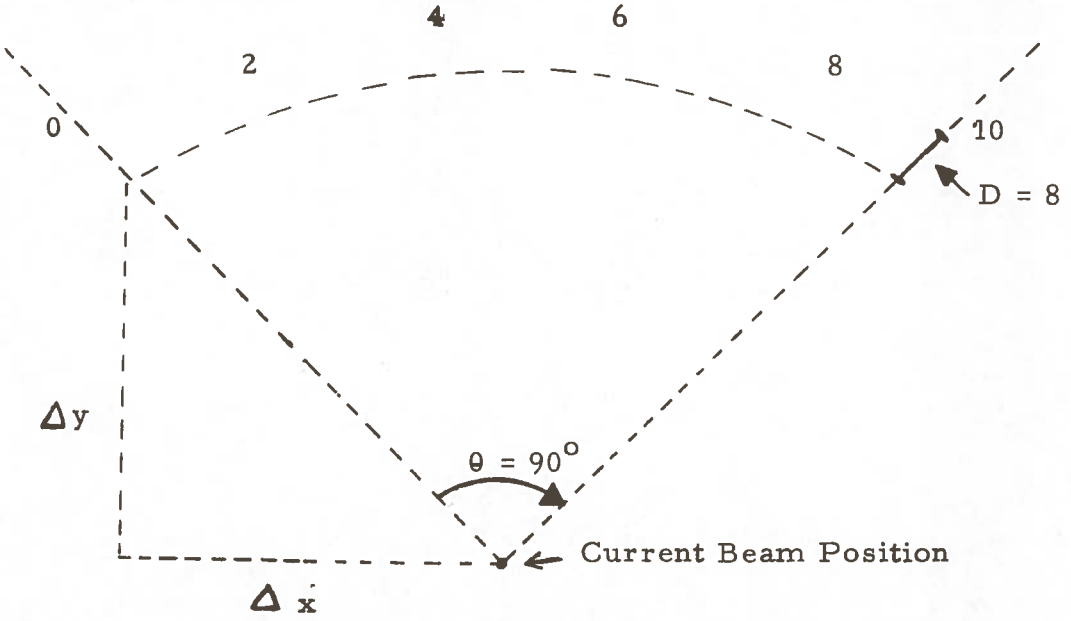
D = distance from arc to scale label along radius

θ = angle of arc in degrees which is to be labelled (If θ is not given, it is assumed to be 360°.)

Numeric labels are displayed around the arc with center point at the current beam position. The arc itself need not be displayed for labels to be drawn. Final beam position remains at the center of the arc.

Example:

:SA Δx, Δy, A, B, N, N, 8, 90



3.5.11 ALPHANUMERICS

:AN S, TEXT

Where S = size of characters (1 - 4). (4 is largest.)

The text material is displayed beginning at the current beam position (X_c, Y_c). Only one line of text is accepted per command. The final beam position (X_f, Y_f) can be calculated as follows:

$(X_f, Y_f) = (N*15*S + X_c, Y_c)$,

Where N = number of characters

3.5.12 DIGITAL READOUT

:DR expr; N, S

Where expr = dynamic expression
N = number of digits to be displayed (maximum of 6)
S = size of characters (1 - 4)

During the create phase N zeroes are displayed starting at the current beam position. At simulation time expr is evaluated and displayed. If N = 6, five digits plus sign are displayed. If N < 6, the specified number of digits will be displayed without sign.

3.6 Control Commands

The control commands may be given at any level of operation. With the exception of Quit, these commands link the create/edit and simulation phases.

3.6.1 SAVE DATA STRUCTURE

:SV Name, Output unit, Phase

Where Name = 6 character name to be saved
Output unit = D disk or
T tape
Phase = C continue in create phase, or
O go to simulate phase

The data structure is saved on disc or tape. The next command given must be at the frame level. If the unit and phase are not specified, the data structure is saved on disk and control goes to the create phase.

3.6.2 RELOAD DATA STRUCTURE

:RL Name, Input unit.

(See SAVE DATA STRUCTURE above for definition of arguments.)

The data structure is reloaded from disc or tape. The next command given must be at the frame level. When no unit is specified, default option is disk.

3.6.3 RESTART

:RS

The data structure is initialized, and previous data structure is deleted from core. The next command given must be the frame command.

3.6.4 QUIT

:Q

Control returns to DOS. Display refresh is halted. The existing data structure is retained in core.

REFERENCES

1. NASA 12-2051, "Programmers Reference Manual for the Dynamic Display Software System", Prepared by Wolf Research and Development Corporation and AZREX, Inc., 1970.

2. Poduska, J. W., et. al., "DOS, The PDP-516 Disk Operating System", United States Government Memorandum, February 1969.

Level	Mnemonic	Description	Command Type	Format	Reference (Page)
Frame	F	Frame	Data Structuring	:F ID	7
Frame	DRE	Define Register	Register Definition	:DRE RXX, expr	9
Frame	D	Delete Frame	Editing	:D F ID	9
Frame	SV	Save Data Structure	Control	:SV Name, Output unit, Phase	23
Frame	RL	Reload Data Structure	Control	:RL Name, Output unit, Phase	24
Frame	RS	Restart	Control	:RS	24
Frame	Q	Quit	Control	:Q	24
Indicator	I	Indicator	Data Structuring	:I ID	7
Indicator	IN	Intensity	Conditioning	:IN I	11
Indicator	TX	Texture	Conditioning	:TX T	11
Indicator	B	Blink	Conditioning	:B	11
Indicator	U	Unblink	Conditioning	:U	11
Indicator	OF	Off	Conditioning	:OF	11
Indicator	ON	On	Conditioning	:ON	12
Indicator	N	New Frame	Conditioning	:N ID	12
Indicator	T	Translate	Conditioning	:T Δx, Δy	12
Indicator	SK	Skip	Conditioning	:SK N	13
Indicator	EQ	Skip if Equal	Conditioning	:EQ A, B, N	13
Indicator	G	Skip if Greater Than	Conditioning	:G A, B, N	14
Indicator	GE	Skip if Greater Than or Equal	Conditioning	:GE A, B, N	14

Level	Mnemonic	Description	Command Type	Format	Reference (Page)
Indicator	LT	Skip if Less Than	Conditioning	:LT A, B, N	14
Indicator	LE	Skip if Less Than or Equal	Conditioning	:LE A, B, N	14
Indicator	STI	Set Initial Texture, Intensity	Drawing	:STI T, I	16
Indicator	D	Delete Indicator	Editing	:D F ID, I ID	9
Indicator	SV	Save Data Structure	Control	:SV Name, Output unit, Phase	23
Indicator	RL	Reload Data Structure	Control	:RL Name, Output unit, Phase	24
Indicator	RS	Restart	Control	:RS	24
Indicator	Q	Quit	Control	:Q	24
Entity	E	Entity	Data Structuring	:E ID	7
Entity	IN	Intensity	Conditioning	:IN I	11
Entity	TX	Texture	Conditioning	:TX T	11
Entity	B	Blink	Conditioning	:B	11
Entity	U	Unblink	Conditioning	:U	11
Entity	OF	Off	Conditioning	:OF	11
Entity	ON	On	Conditioning	:ON	12
Entity	N	New Frame	Conditioning	:N ID	12
Entity	T	Translate	Dynamic Conditioning	:T x-expr; y-expr	12
Entity	R	Rotate	Conditioning	:R x-expr; y-expr; e-expr	13

Level	Mnemonic	Description	Command Type	Format	Reference (Page)
Entity	SK	Skip	Conditioning	:SK N	13
Entity	EQ	Skip if Equal	Conditioning	:EQ A, B, N	13
Entity	G	Skip if Greater Than	Conditioning	:G A, B, N	14
Entity	GE	Skip if Greater Than or Equal	Conditioning	:GE A, B, N	14
Entity	LT	Skip if Less Than	Conditioning	:LT A, B, N	14
Entity	LE	Skip if Less Than or Equal	Conditioning	:LE A, B, N	14
Entity	S	Set Point	Drawing	:S X, Y	17
Entity	P	Point	Drawing/Dynamic Drawing	:P Δx #x-expr; Δy #y-expr	17
Entity	BL	Blank Line	Drawing/Dynamic Drawing	:BL Δx #x-expr; Δy #y-expr	18
Entity	L	Line	Drawing/Dynamic Drawing	:L Δx #x-expr; Δy #y-expr	18
Entity	A	Arc	Drawing	:A Δx, Δy, e	18
Entity	AN	Alphanumerics	Drawing	:AN S, Text	23
Entity	DR	Digital Readout	Dynamic Drawing	:DR expr; N, S	23
Entity	TM	Tick Marks for Lines	Drawing	:TM N, X_i, Y_i, X_f, Y_f, Δx, Δy	19
Entity	SL	Scales for Lines	Drawing	:SL X_i, Y_i, X_f, Y_f, A, B, N, Size	21
Entity	TMA	Tick Marks for Arcs	Drawing	:TM Δx, Δy, N, L, e	20
Entity	SA	Scales for Arcs	Drawing	:SA Δx, Δy, A, B, N, Size, D, e	22

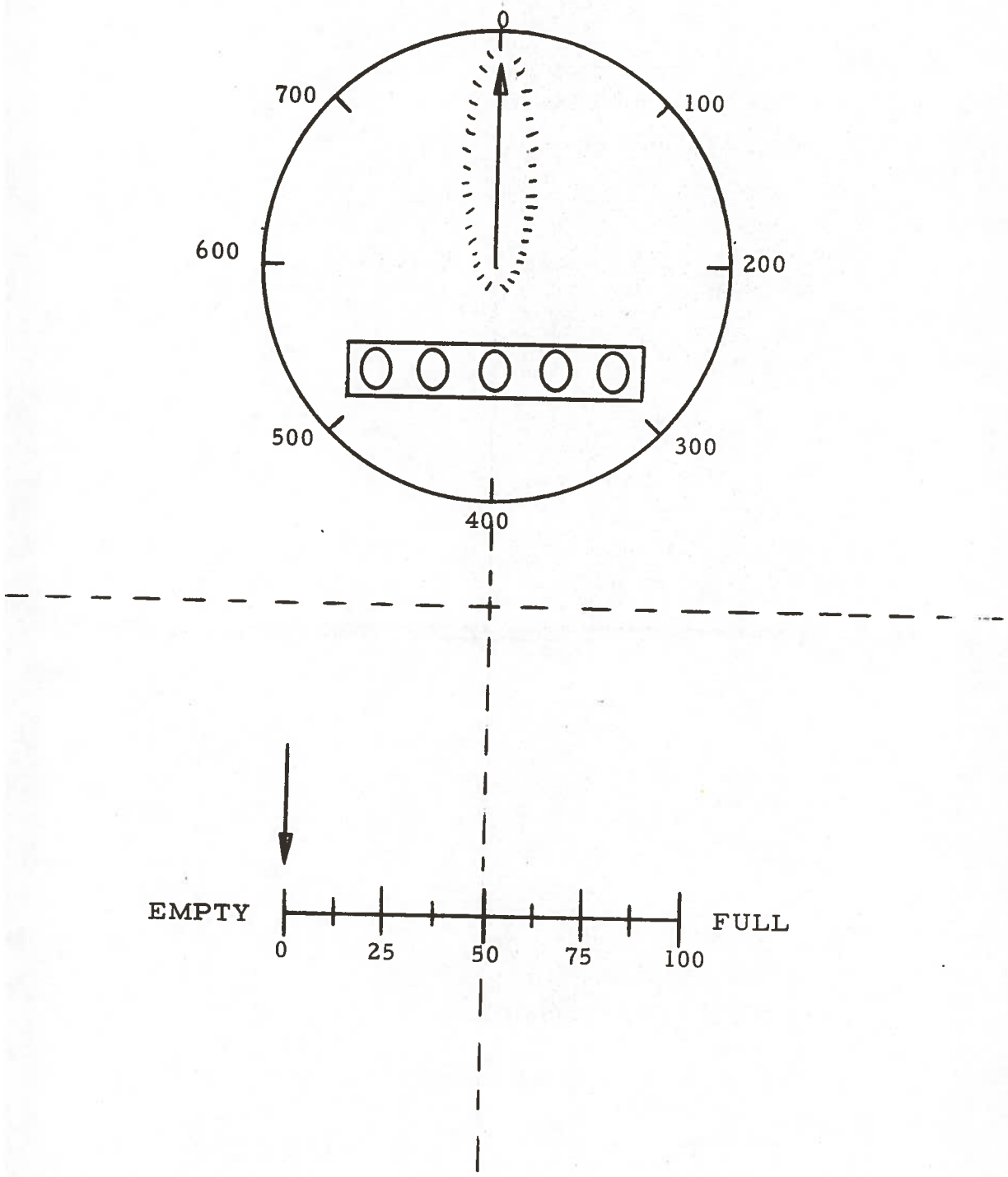
Level	Mnemonic	Description	Command Type	Format	Reference (Page)
Entity	STI	Set Initial Texture, Intensity	Drawing	:STI T, I	16
Entity	D	Delete Entity	Editing	:D F ID, I ID, E ID	9
Entity	D	Delete Current Entity	Editing	:D	9
Entity	SV	Save Data Structure	Control	:SV Name, Output unit, Phase	23
Entity	RL	Reload Data Structure	Control	:RL Name, Output unit, Phase	24
Entity	RS	Restart	Control	:RS	24
Entity	Q	Quit	Control	:Q	24

Mnemonic	Description	Format	Command Type	Level	Reference (page)
A	Arc	:A Δx, Δy, θ	Drawing	Entity	18
AN	Alphanumerics	:AN S, Text	Drawing	Entity	23
B	Blink	:B	Conditioning	Indicator or Entity	11
BL	Blank Line	:BL Δx# x-expr;Δy #y-expr	Drawing or Dynamic Drawing	Entity	18
D	Delete Frame	:D F ID	Editing	Frame	9
	Delete Indicator	:D F ID, I ID		Indicator	9
	Delete Entity	:D F ID, I ID, E ID		Entity	9
	Delete Current Entity	:D		Entity	9
DR	Digital Readout	:DR expr; N, S	Dynamic Drawing	Entity	23
DRE	Define Register	:DRE RXX, expr	Register Definition	Frame	9
E	Entity	:E ID	Data Structuring	Entity	7
EQ	Skip if Equal	:EQ A, B, N	Conditioning	Indicator or Entity	13
F	Frame	:F ID	Data Structuring	Frame	7
G	Skip if Greater Than	:G A, B, N	Conditioning	Indicator or Entity	14
GE	Skip if Greater Than or Equal	:GE A, B, N	Conditioning	Indicator or Entity	14
I	Indicator	:I ID	Data Structuring	Indicator	7
IN	Intensity	:IN I	Conditioning	Indicator or Entity	11

Mnemonic	Description	Format	Command Type	Level	Reference (page)
L	Line	:L Δx #x-expr ; Δy #y-expr	Drawing or Dynamic Drawing	Entity	18
LE	Skip if Less Than or Equal	:LE A, B, N	Conditioning	Indicator or Entity	14
LT	Skip if Less Than	:LT A, B, N	Conditioning	Indicator or Entity	14
N	New Frame	:N ID	Conditioning	Indicator or Entity	12
OF	Off	:OF	Conditioning	Indicator or Entity	11
ON	On	:ON	Conditioning	Indicator or Entity	12
P	Point	:P Δx #x-expr; Δy #y-expr	Drawing or Dynamic Drawing	Entity	17
Q	Quit	:Q	Control	Frame, Indicator, or Entity	24
R	Rotate	:R x-expr; y-expr; ɵ-expr	Dynamic Conditioning	Entity	13
RL	Reload Data Structure	:RL Name, Output unit, phase	Control	Frame, Indicator, or Entity	24
RS	Restart	:RS	Control	Frame, Indicator, or Entity	24

Mnemonic	Description	Format	Command Type	Level	Reference (page)
S	Set Point	:S X, Y	Drawing	Entity	17
SA	Scales for Arcs	:SA $\Delta x, \Delta y$, A, B, N, Size, D, Θ	Drawing	Entity	22
SK	Skip	:SK N	Conditioning	Indicator or Entity	13
SL	Scales for Lines	:SL X_i, Y_i, X_f, Y_f, A, B, N, Size	Drawing	Entity	21
STI	Set Initial Texture, Intensity	:STI T, I or :STI I, T	Drawing	Indicator or Entity	16
SV	Save Data Structure	:SV Name, Output Unit, phase	Control	Frame, Indicator, or Entity	23
T	Translate	:T $\Delta x, \Delta y$:T x-expr; y-expr	Conditioning Dynamic Conditioning	Indicator Entity	12
TM	Tick Marks for Lines	:TM N, X_i, Y_i, X_f, Y_f, $\Delta x, \Delta y$	Drawing	Entity	19
TMA	Tick Marks for Arcs	:TMA $\Delta x, \Delta y$, N, L, Θ	Drawing	Entity	20
TX	Texture	:TX T	Conditioning	Indicator or Entity	11
U	Unblink	:U	Conditioning	Indicator or Entity	11

APPENDIX C - EXAMPLE



```
OK                                  [Underlined words indicate DOS response]
ATTACH WRIGHT                       [User File Directory Name]
OK
R DISSYS                            [Restore Create/Edit Phase]
GO
```

```
:F 1
:DRE R10, 650
:DRE R11, 25
:I 1                    [INDICATOR 1: AIR SPEED INDICATOR]

:E 1                                [Arc, tick marks, and scales]
:S 0, 250
:A 200
:TMA 0, 192, 8, 16
:SA 0, 200, 0, 700, 8, 1, 10

:E 2                                [Pointer to Speed]
:S 0, 250
:R 0, 250, R8*(360/800)             [R8 = Current Air Speed]
:L 0, 180
:L 5, -10
:L -10, 0
:L 5, 10
:G R8, R10, 1                       [Blink if air speed exceeds 650 mph]
:SK 1
:B

I 1
STI S,3
```

```
:I 2                    [INDICATOR 2: ALTIMETER]

:E 1                                [Altimeter rectangular boundary]
:S -75, 150
:L 150,0
:L 0, 40
:L -150, 0
:L 0, -40

:E 2                                [Altimeter read-out]
:S -70, 160
:DR R3, 5, 2                        [R3=Distance from ground zero]
:STI S
```

```
:I 3
```

[INDICATOR 3: FUEL GAUGE]

```
:E 1
:S -160, -200
:L 320, 0
:BL -320, 0
:TM 5, 0, 0, 320, 0, 0, 20
:BL -320, 0
:TM 4, 40, 0, 280, 0, 0, 10
```

[Line and Tick Marks]

```
:E 2
:S -160, -220
:SL 0, 0, 320, 0, 0, 100, 5, 1
:BL -460, 0
:AN 2, EMPTY
:BL 370, 0
:AN 2, FULL
```

[Scales and Labels]

```
:E 3
:S -160, -100
:L 0, -50
:L 5, 7
:L -10, 0
:L 5, -7
:T R5*320/100, 0
:LT R11, R5, 1
B
I 3
STI S, 3
SV XAMPLE, D, O
DATA = XAMPLE DT = 100
POLL DG = N, PPI = N, PDP = N, PCS = N
```

[Pointer]

[R5=Percent Fill of Fuel Tank]

[Simulation Phase Commands
Underlined characters indicate monitor output.]