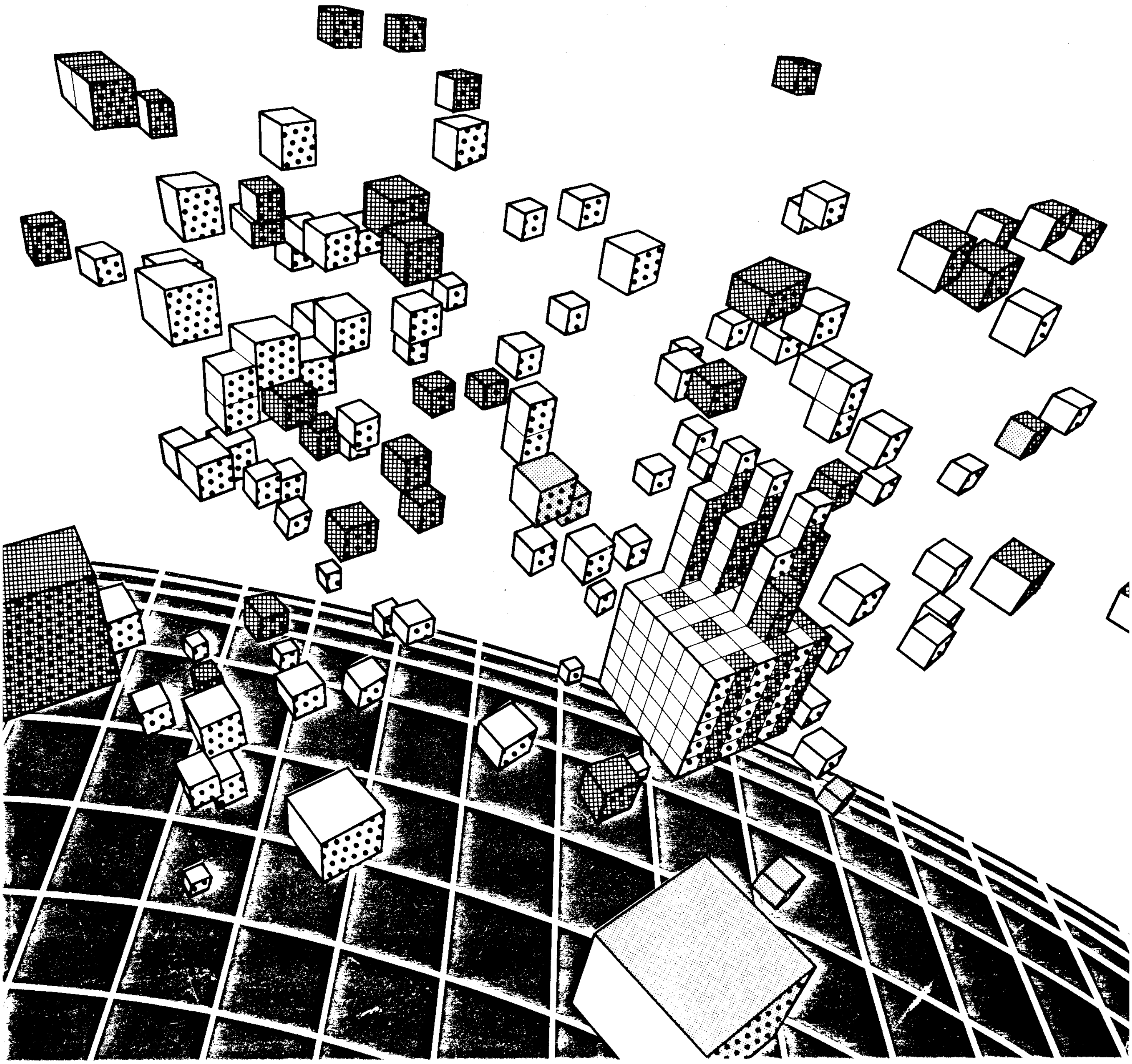
# InterGraphics '83

## Tutorial Sessions A 5



This Page Intentionally Left Blank

DATAMAX UV-1
Zgrass GRAPHICS SYSTEM

# Zgrass TRANS PACKAGE
# DOCUMENTATION & GLOSSARY

Software Tools for
Three Dimensional Vector Images

January 12, 1983

Real Time Design, Inc.
531 South Plymouth Court, Suite 102
Chicago, Illinois 60605

## COPYRIGHT NOTICE

## DISCLAIMER

Real Time Design, Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantibility or fitness for any particular purpose. Further, Real Time Design, Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Real Time Design, Inc. to notify any person or organization of such revision or changes.

## GETTING STARTED WITH TRANS

Before you start your first session, refer to the **Zgrass Utility Disk User's Guide** for instructions on how to make a copy of the TRANS disk. After you've made a copy of the disk, insert the copy into drive 0 of your disk drive and type:

```
DS 0;DG GETRANS;GETRANS
```

Online help is available with the GETHELP and COMMANDS macros. Also, the GETDEMO macro provides a menu of sample macros which DEMOnstrate system features. You can examine these examples of TRANS command sequences by printing the DEMO macros:

```
DG DOPYR;DOPYR;PR DOPYR
```

A categorized summary and Glossary of the TRANS commands are included in this guide.

## INTRODUCTION

The Zgrass TRANS PACKAGE, generally referred to as TRANS for short, is a set of macros and swap modules designed for modelling three-dimensional vector (wire-frame) objects on the Datamax UV-1 Zgrass Graphics System. It is targeted for people who are familiar with the basic programming concepts of the Zgrass language, as well as three-dimensional coordinates, and geometric transformations.

The basic approach to TRANS involves three steps:

o <u>Object creation</u>: objects are defined by storing the values of X, Y, Z, and COLORMODE in 2-dimensional arrays. The macro GETOBJ offers a menu of macros which can be used to define wire-frame objects.

o <u>Transformation definition</u>: geometric transformations are defined through the use of commands such as HOME, MOVE, ROT, SCALE3D, and SHEAR. Successive operations are multiplied together and saved in a transformation array.

o <u>Image display</u>: images are displayed by (optionally) applying the transformation array to the coordinates as the vectors are drawn on the screen with the commands TDISP and ODISP. Simple perspective can be used to clarify the three-dimensional nature of the images with the PDISP command.

Screen animations are easily created with incremental transformations displayed on different screens. These are played back by 'flipping' through the screens in sequence with the FLIP command. The DEMO macros illustrate this technique.

## USING THE TRANS PACKAGE

Although some menus are provided to get you started, the TRANS PACKAGE is basically a command driven system. In TRANS, only one object is currently active, or OPEN, at a time. All transformation and display commands apply to the current OPENed object. Several objects can be GROUPed together in a hierarchical (tree) structure. GROUPs can be OPENed and then transformed with the same commands as simple objects.

TRANS commands can be utilized in two ways. The first way, command mode, is by entering commands separately on the terminal keyboard. The second way is to create a macro with the desired sequence of commands. See the following examples:

```
.EXAMPLE1 command mode
DGET SETPRISM
SETPRISM ETTA,100,100,100; .create a prism called ETTA
OPEN ETTA;. make ETTA the current object
ROT Y,40;. rotate 40 degrees around Y axis
ROT X,12;. rotate 12 degrees around X axis
CLEAR
TDISP;. display
HOME;. reset transformations
MOVE 90,20,0;. move to the right & up
TDISP
SHOWALL;. list the coordinates on the terminal


EXAMPLE2=[. macro mode
DGET SETCUBE
SETCUBE CHRISTINE,100,100,100;. create a cube
OPEN CHRISTINE;. make CHRISTINE the current object
SCALE3D 2,1,1;. scale in X direction
CLEAR
PDISP 200;. display with perspective
SCALE3D .5,.5,.5;. scale in all directions
PDISP 200
]
EXAMPLE2
```

## PROGRAMMING NOTES

### Array types and data structures

The primary data structures of the TRANS PACKAGE are vector lists, transformation matrices, and groups:

- o Vector lists are Nx4 structures which can be stored in either INTEGER or FLOATING POINT arrays. The primary advantage of INTEGER arrays is that they require approximately one half of the memory that FLOATING POINT arrays require for the same number of points. However, INTEGER arrays may introduce more roundoff error than FLOATING POINT arrays. This will be a problem particularly if the REFORM command is used, since INTEGER arrays will cause the new coordinate values to be truncated to integers. In general, INTEGER arrays are sufficient for uses where the screen coordinate system can also be used as data coordinates.

- o Transformation matrices are 4x4 arrays. They should always be declared as FLOATING POINT since compounding of transformations generally results in floating point values.

- o Groups are used to create hierarchical (tree) structures. Groups are defined as one dimensional STRING arrays. Each element of a group is a NAME, either the name of a vector list or the name of another group. See the TRANS Glossary entries for the SGET and SPUT swap commands to see how to store and retrieve string arrays from disk.

A TRANS structure consists of two parts - a data array (vector list or group) and its companion transformation matrix. The characters T$ are prefixed to the data array name to derive the name for the matrix. See the TRANS Glossary entry on NAMES for further details.

## Space considerations

If you are processing lots of separate vector lists or dealing with objects with many vectors you may find that you run out of space in memory. The following are a few tips on how to reduce memory requirements.

You can delete some of the TRANS PACKAGE macros which you do not need without interferring with performance of other macros. These can be retrieved from disk with the DGET command if wanted later. The first likely candidates are the GETOBJ, GETDEMO, and GETHELP macros. These are mainly needed for reference when learning the system and are not used by any other macros. The COMMANDS macro can also be deleted without harm, but it is a useful reference even for an experienced TRANS user. If you are not using groups, the GROUP, GHOME and GDISP macros can be eliminated. The FLIP macro is only used for screen animations, ND.B is a quick alternative (it is a resident system command).

If your normal use of the TRANS PACKAGE suffers from space problems, the GETRANS macro can easily be c istomized so unwanted macros are not loaded into memory. It is advised that you comment out unwanted lines instead of deleting them. This can be done by editing GETRANS and making the lines which DGET the unwanted macros into comments by inserting a . (period) as the first character of the line. You may wich to save the updated GETRANS on your TRANS disk.

## The coordinate system

TRANS uses a standard X,Y,Z coordinate system with 0,0,0 being at the center of the screen. Positive X values are to the right and positive Y values are up. The TRANS PACKAGE uses a right-handed coordinate system with the positive Z axis coming out of the screen toward the viewer. Positive rotations about an axis are counterclockwise. You can use the following list as a reference for rotational direction:

A 90 degree rotation about the X axis moves +Y to +Z.
A 90 degree rotation about the Y axis moves +Z to +X.
A 90 degree rotation about the Z axis moves +X to +Y.

Single point perspective views are generated from a vantage point somewhere on the Z axis. The location on the axis is thought of as "pixels away from the screen".

## Creating objects

The GETOBJ menu provides a list of macros which create TRANS objects. Additional macros to create vector lists can easily be written. Any of the object creation macros, such as SETCUBE, will serve as a good example of the type of commands needed. The only special step which must be done is to use the **ARRDATA** swap command to keep track of the number of vectors in the list. For example:

```
.EXAMPLE3 object creation
ARRAY.I TRI,9,4;. space for 9 points
OPEN TRI
SETPOINT 0,50,-20,0,4;. SETPOINT uses OPEN array
SETPOINT 1,0,50,0,1
SETPOINT 2,-50,-20,0,2
SETPOINT 3,50,-20,0,3
ARRDATA TRI,3,4;. number of points used
MOVE -60,0,0;. move to left
CLEAR
TDISP;. display
SETPOINT 4,0,50,0,4
SETPOINT 5,0,-80,0,3
ARRDATA TRI,3,6;. number of points used
MOVE 120,0,0;. move to right
TDISP
```

## Summary of TRANS Commands

* General *

```
OPEN arrayname
COPY from,to
PARTCOPY from,to,start,stop
SHOWALL
REFORM
GROUP groupname,count,name,name,...
MERGE finalname,type,count,name,name,...
```

* Transformations *

```
HOME
MOVE x,y,z
ROT axis,angle
SCALE3D x,y,z
SHEAR 1axis,2axis,value
GHOME
```

* Help Menus *

```
GETHELP;. help
GETOBJ;.  object creation macros
GETDEMO;. demo macros
```

* Display [optional argument] *

```
ODISP [colormode]
PDISP distance[,colormode]
TDISP [colormode]
GDISP gname
```

* Animation *

```
FLIP lowscreen,highscreen,delay
```

```
ARRCAT DESTARRAY,ARRAY1,LOWER1,UPPER1,...,ARRAYn,LOWERn,UPPERn
```

Esoteric Swap Command

takes the values ARRAYi(LOWERi) thru ARRAYi(UPPERi), for arrays 1 to n, concatenates them together, and creates the new array DESTARRAY. The arrays may be of any type (FLOAT, INTEGER, STRING), as long as all of them are of the same type. DESTARRAY is always created as a one dimensional array.

```
ARRDATA ANAME,1
ARRDATA ANAME,2
ARRDATA ANAME,3,NUMBER
ARRDATA ANAME,4
ARRDATA ANAME,5
ARRDATA ANAME,6,NUMBER
```

Esoteric Swap Command

stores and retrieves information in the Zgrass system header of array ANAME. The second argument is a flag indicating which information to process. The third argument (when needed) is used to update the header. One important application of ARRDATA is to maintain a counter stored in the header. WARNING!! ARRDATA should be used with one or two dimensional arrays only. (We call the first dimension rows and the second columns.) ARRDATA acts according to the following table:

| Flag | Action |
|---|---|
| 1 | RETURNs the number of rows |
| 2 | RETURNs the current value of the counter |
| 3 | stores NUMBER in the counter |
| 4 | RETURNsa code for the array element type (5=INTEGER, 7=FLOATING POINT, 9=STRING) |
| 5 | RETURNs the number of columns |
| 6 | reconfigures array as a two dimensional array with NUMBER columns |

Example:

```
TEST=[ARRAY PETE,4,3
ARRDATA PETE,3,0;. set counter to 0
PR ARRDATA(PETE,1),ARRDATA(PETE,5),'row column'
PR ARRDATA(PETE,2),'is the counter'
PR ARRDATA(PETE,4),'is the type of element'
ARRDATA PETE,3,2;. set counter to 2
ARRDATA PETE,6,4;. make PETE a 3x4 array
PR ARRDATA(PETE,1),ARRDATA(PETE,5),'row column'
PR ARRDATA(PETE,2),'is the counter'
]
TEST
```

COMD3D
COMG3D1
COMG3D2
COMO3D
COMT3D
Filename

these are a series of files which are used by the GETHELP macro to provide online help.

COMMANDS
Trans Command

will show a brief index of TRANS commands.

COPY SOURCE,DESTINATION
Trans Command

copies array SOURCE to array DESTINATION. COPY can be used to copy transformation arrays as well as vector lists.

DEBUG
Idiosyncracy

if DEBUG=1 before running GETRANS, none of the macros will be compiled. This is useful when modifying or examining the TRANS macros.

DEMO
Idiosyncracy

macros provided on the TRANS disk to demonstrate the system. Any of the following can be printed by the user as illustrations of command sequences DOCUBE, DOGRID, DOGROUP, DOPRISM, DOPYR, DOQUBE.

DOCUBE
Trans Command

SETCUBE DEMOnstration macro. Also demonstrates effect of changing perspective distance with PDISP for successive image views.

DOGRID
Trans Command

SETGRID and SHEAR DEMOnstration macro.

DOGROUP
Trans Command

DEMOnstrates the use of GROUPs for modelling and displaying hierarchical structures.

DOPRISM
Trans Command

SETPRISM DEMOnstration macro. Also uses SCALE3D, MOVE, and the color override optional argument of PDISP.

DOPYR
Trans Command

ROT and SETPYRAMID DEMOnstration macro.

DOQUBE
Trans Command

7 rotating cubes DEMOnstration macro. This macro uses COPY, MERGE and REFORM to construct a single large vector list before displaying it.

DODISP ARRAY
DODISP ARRAY,COLORMODE
DODISP ARRAY,START,STOP
DODISP ARRAY,START,STOP,COLORMODE
Trans Swap

Displays vectors using the elements of a Nx4 ARRAY as X,Y,Z, and COLORMODE values. The vectors are displayed with an orthogonal projection. If COLORMODE is given, it overrides all stored colormodes except 4. The START and STOP arguments are used to draw a subsection of an ARRAY.

DPDISP ARRAY,TARRAY,DISTANCE
DPDISP ARRAY,TARRAY,DISTANCE,COLORMODE
DPDISP ARRAY,TARRAY,DISTANCE,START,STOP
DPDISP ARRAY,TARRAY,DISTANCE,START,STOP,COLORMODE
Trans Swap

displays vector using the elements of an Nx4 ARRAY as X,Y,Z, and COLORMODE values. Each point is multiplied by TARRAY, a 4x4 transformation array, before being displayed. A perspective view is generated as DISTANCE "pixels from the screen". If COLORMODE is given, stored colormodes except 4 are overridden. The START and STOP arguments are used to draw a subsection of an array. Note: weird effects may occur if some Z coordinates are larger than DISTANCE.

DTDISP ARRAY,TARRAY
DTDISP ARRAY,TARRAY,COLORMODE
DTDISP ARRAY,TARRAY,START,STOP
DTDISP ARRAY,TARRAY,START,STOP,COLORMODE
Trans Swap

displays vectors using the elements of an Nx4 ARRAY as X,Y,Z, and COLORMODE values. Each point is multiplied by TARRAY, a 4x4 transformation array, before being displayed. If COLORMODE is given, stored colormodes except 4 are overridden. The START and STOP arguments are used to draw a subsection of an array.

EDARRAY
Trans Command

converts the OPENed array into the string $Edarray and invokes EDIT so one can insert, delete, etc. Use RESTORE to convert $Edarray back into an array. This command is not loaded by GETRANS.

FLIP LOWSCREEN,HISCREEN,DELAY
Trans Command

does a multi-screen animation by changing $TV from LOWSCREEN to HISCREEN and then back to LOWSCREEN. DELAY is used to control the rate of change. Each DEMOnstration macro uses FLIP.

GDISP GNAME
Trans command

displays GROUP GNAME by displaying each member in turn. To display members which are themselves GROUPs, GDISP calls itself recursively. A global array, $Ctm, is used to pass compounded transformations to the successive members of a GROUP.

GETDEMO
Trans Command

provides a menu listing the TRANS PACKAGE DEMOnstration macros on the terminal. It gets the chosen macro from disk and runs it.

GETHELP
Trans Command

provides a menu listing the TRANS PACKAGE help options. The selected option is read from disk and displayed on the terminal.

GETOBJ
Trans Command

provides a menu of object generation macros. The selected macro(s) are read from disk.

GETRANS
Trans Command

loads TRANS PACKAGE from disk.

GHOME
Trans command

sets the global current transformation matrix, $Ctm, to the identity matrix. This matrix is used by GDISP when displaying groups.

GROUP GROUPNAME,COUNT,NAME,NAME,NAME,...
Trans command

creates a group called GNAME with COUNT members; NAME, NAME,... The members can be vector lists or other GROUPs. GROUPs are used for dealing with hierarchical collections of objects and allow compounding multiple levels of transformations. Use GDISP to display GROUPs. See the macro DOGROUP for an example which uses groups.

HOME

Trans Command

sets the OPENed object's transformation matrix to identity.

INTRO3D

Filename

introduction to TRANS. Can be accessed with GETHELP.

MERGE FINALNAME,TYPE,COUNT,NAME,NAME,NAME...

Trans Command

merges COUNT number of objects into a new array called FINALNAME. If TYPE is 1, each object's transformations are applied to its coordinates when merging. If TYPE is 0, the coordinates are merged unchanged.

MOVE X,Y,Z

Trans Command

moves the OPENed object X,Y,Z pixels in the X, Y,and Z directions by multiplying the proper matrix into the transformation array.

MXMUL ARRAY,ARRAY,DEST

Swap Command

multiplies a MxN array with a NxP array using the rules for matrix multiplication. The resultant MxP array is stored as DESTARRAY.

NAMES

Idiosyncrasy

TRANS PACKAGE names and global variables. Global variables have names which begin with the $ character.

These 4 variables are given values by the OPEN command:

| | |
|---|---|
| $Array | numeric array storing X,Y,Z and COLORMODE values or string array storing GROUP member names |
| $Matrix | FLOATING POINT array storing 4x4 geometric transformation matrix |
| $Araynam | character string with name for OPENed object |
| $Mtrxnam | character string with name for OPENed matrix |

The prefix T$ in an array name indicates that the array is a 4x4 transformation array (i.e., T$PIX is the matrix for PIX). The OPEN command creates an array with such a name if one does not exist when an object is OPENed.

$Ctm is a global transformation matrix used by the GDISP command when displaying GROUPs.

$Edarray is a string with X,Y,Z,COLORMODE values on each line, separated by commas but no spaces. This string is created by the EDARRAY command; it can also be created with EDIT. The RESTORE command turns $Edarray into a coordinate array.

ODISP
ODISP COLORMODE
Command

displays the OPENed object without transformations. If COLORMODE is specified, it overrides the COLORMODEs stored in the vector list.

OPEN ARRAYNAME
Command

makes ARRAYNAME the object to which transformation, display, and other commands will apply. OPEN creates a transformation matrix (named 'T$'&ARRAYNAME) for the object if one doesn't exist. See NAMES for more info.

PARTCOPY SOURCE,DESTINATION,START,STOP
Trans Command

copies row START through row STOP from array SOURCE to array DESTINATION.

PDISP DISTANCE
PDISP DISTANCE,COLORMODE
Trans Command

displays the OPENed object with its transformations and a perspective projection viewed DISTANCE "pixels from the screen". A right-handed coordinate system is used, with +Z coming out of the screen. If COLORMODE is specified, it overrides the COLORMODEs stored in the vector list.

REFORM
Trans Command

makes new object coordinates by multiplying the transformation matrix into the vector list of the OPEN object.

RESTORE NAME
Trans Command

converts string $Edarray into an array called NAME with proper length and OPENs NAME. Commonly used after EDARRAY, but can be used independently, provided $Edarray has the correct format. $Edarray must have values for X,Y,Z, and COLORMODE on each line, separated by commas without spaces, for RESTORE to work properly. RESTORE is not gotten from the disk by GETRANS. RESTORE uses the swap SFIND.

ROT AXIS,ANGLE
Trans Command

rotates the OPENed object ANGLE degrees about the specified AXIS by multiplying the proper rotation matrix into the transformation array. AXIS is X,Y, or Z.

SCALE3D X,Y,Z
Trans Command

scales the OPENed object by factors of X,Y,Z along the X, Y, and Z axes. SCALE3D multiplies the proper scale matrix into the transformation array.

SETCUBE NAME,WIDTH,HEIGHT,DEPTH
Trans Command

creates an INTEGER array of cube coordinates with dimensions WIDTH,HEIGHT,DEPTH. The array, called NAME, is OPENed but is not displayed by this command. Can be accessed with the GETOBJ menu.

SETGRID NAME,XSIZE,YSIZE,INCREMENT,COLORMODE
Trans Command

creates a plane grid of dimensions XSIZE by YSIZE. Vectors are given COLORMODE and spaced INCREMENT pixels apart. Can be accessed with the GETOBJ menu.

SETKEY NAME,SIZE
Trans Command

used to enter SIZE coordinates into array NAME with the terminal keyboard. Included in the GETOBJ menu.

SETPOINT INDEX,X,Y,Z,COLORMODE
Trans Command

assigns X,Y,Z, and COLORMODE values to row INDEX of the OPENed array. This command is useful when defining objects.

SETPRISM NAME,BASE,HEIGHT,DEPTH
Trans Command

creates an INTEGER array with the coordinates of a rectangular prism of base BASE, height HEIGHT, and depth DEPTH. The array (NAME) is OPENed, but is not displayed. Can be accessed with the GETOBJ menu.

SETPYRAMID NAME,BASE,HEIGHT
Trans Command

creates an INTEGER array with the coordinates of a square pyramid of base BASE and height HEIGHT. Can be accessed with the GETOBJ menu.

SETABLET NAME,SIZE
Trans Command

creates array NAME with space for SIZE coordinates. Vectors are entered into the array by drawing with the cursor on the tablet. All Z coordinates are stored as 0. Can be accessed with the GETOBJ menu.

SFIND STRING,START,CHAR
Esoteric Swap Command

returns the location of CHAR in the string STRING. Searching begins at location START in the string. Use START=0 to search the entire string. If CHAR is not found, SFIND returns -1. Note that CHAR must be given as an INTEGER, not as a character string. See the ASCII table in the GLOSSARY for character codes. Example:

```
TERRY=[ABCDABC]
PR SFIND(TERRY,0,65),SFIND(TERRY,1,65),ASCII(65)
```

will print

```
0 4 A
```

SGET NAME
Esoteric Swap Command

copies the string array from the current screen back into user memory as a regular string array. This command is used to restore a string array which has been stored on disk using the SPUT and DPUT.S commands. See SPUT for an example using this command.

SHEAR 1AXIS,2AXIS,VALUE
Trans Command

shears the OPENed object on 1AXIS along 2AXIS by VALUE. 1AXIS and 2AXIS must be either X, Y, or Z.

SHOWALL
Command

prints the transformation matrix and the vector list of the OPENed object on the terminal. If a GROUP is OPEN, the members are listed.

SPUT(STRINGARRAY)
Esoteric Swap Function

takes the STRINGARRAY and stores it in the current screen. SPUT returns how many bytes of screen memory were needed to store the array. Note that this swap affects the image on the screen that $MW is set to. Example:

```
ARRAY.STR PATTI,3
PATTI(0)="BANANA"
PATTI(2)="MANGO"
DPUT.S SPUT(PATTI),SPATTI,[STRING ARRAY PATTI]
```

will place the string array PATTI into screen memory and then store it on disk using the name SPATTI. The disk name used in the DPUT command must be different than the name of a string array which is currently in user memory. To get a saved string array off of a disk use the DGET and SGET commands as in:

```
DGET SPATTI
SGET KANGA
PR KANGA(0),KANGA(2)
```

will print

```
BANANA MANGO
```

SYSTEST
Trans Command

this command tests the TRANS PACKAGE by running each of the DEMO macros consecutively.

T$
Idiosyncracy

is the prefix used to identify transformation matrices. For example, T$PIX is the matrix for the array PIX. See NAMES for further information.

TDISP
TDISP COLORMODE
Trans Command

displays the OPENed object with its transformations applied. If COLORMODE is specified, it overrides the COLORMODEs stored in the vector list.

TRANO
TRAN1
TRAN2
Filename

disk files with the macro definitions of COPY, HOME, MOVE, OPEN, PARTCOPY, REFORM, ROT, SCALE3D, SETPOINT, SHEAR, SHOWALL, TDISP, PDISP, ODISP, They are used by GETRANS when initializing the system.

UNITY ARRAY
Trans Swap

sets a 4x4 ARRAY to the identity matrix; i.e., 1's on the main diagonal and 0's elsewhere.

2033 M Street N.W. Suite 399
Washington, D.C. 20036, U.S.A.

Printed by: SEINO PRINTED Co., Ltd.
Tokyo, Japan