TABLE OF CONTENTS

1 Introduction....1

2 Types of THESEUS Routines....3

3 The STARTQ13 Global Data Area....6
3.1 DIM, DIM2, DIM3, SD....7
3.2 BUFPTR, DIRPTR....8
3.3 OPOBJ, OBJDIMN, OBJDIMC....8
3.4 THEFLAGS and Associated EQU's....9
3.5 ATTRPTR, OPOBJPTR, CALLTPTR, VTRANPTR....10
3.6 CALLTYPE and Associated EQU's....11
3.7 DEFATTR....12
3.8 POLLINDX....12
3.9 VIEWDIM and Associated EQU's....12
3.10 CALLDIM and Associated EQU's....13
3.11 CALRBASE, ERRADDR....13
3.12 CURENABL and Associated EQU's....14
3.13 SCREEN, WORLD....14
3.14 MINFIX, MAXFIX....15
3.15 IDMATR2D, IDMATR3D....16
3.16 DDIMLIST, QDYNLIST, ADDMLIST, ADDSLIST, OPENLIST....16
3.17 PROCRUSTES Parameters....17

4 Error Handling....18

5 EXEC's....19
5.1 GENTHE....19
5.2 THEASM....20
5.3 THELINK....20

6 Libraries....21
6.1 Macro Libraries....21
6.1.1 AWXALL, AWXM4A....21
6.1.2 PROCRUST....22
6.1.3 AWXRUNT....22
6.2 Text Libraries....23
6.2.1 PROCRUST....23
6.2.2 THESEUS....24

7 Dynamos....26

8 Callers of PROCRUSTES Routines....28

9 Callers of Internal Routines....29

10 References....30

1 INTRODUCTION

THESEUS is a collection of ALGOLW-callable assembler routines that act as an interface between the ALGOLW programmer and the PROCRUSTES subroutine package (see [2]).[1] Essentially, THESEUS provides the ALGOLW programmer with the facilities of the SIMALE Standard Graphics Package without requiring him to have an understanding of the internals of the graphics package or of BUGS software in general; THESEUS also does extensive error-checking and provides higher-level constructs; THESEUS will appear to the User[2] as an extension of ALGOLW making interactive graphics possible.

The User of THESEUS need only concern himself with [4]. This document is intended to provide an overview of the internal structure of the subroutine package for those interested in [comleting,] maintaining or modifying it. A knowledge of [1], [2], and [3], as well as [4] is assumed.

Basically, the routines comprising THESEUS are short and simple. The only difficulties are encountered as a result of the following factors:

---

[1] PROCRUSTES in turn acts as an interface between the assembler programmer and the SIMALE Standard Graphics Package

[2] Whenever referred to in this document, the "User" is an ALGOLW programmer writing and executing a graphics program for BUGS.

- The package is one level of a hierarchy of software (from bottom to top: SSGP, PROCRUSTES, THESEUS, User's ALGOLW program, ALGOLW runtime environment), and a certain amount of familiarity with all of these levels is required -- worse, the distinction between THESEUS and the levels immediately above and below it is sometimes blurred because it is sometimes necessary for THESEUS to muck with the internals of other levels.

- Many THESEUS routines can be passed another routine as a parameter. This passed procedure can be another THESEUS routine, or it can consist of a series of calls to several THESEUS routines. This causes two problems: (1) certain data structutes must be made "global" so that those procedures passed to THESEUS as parameters can access them, and (2) certain THESEUS routines may not be passed as parameters, while the others may only be invoked as parameters passed to other THESEUS routines and then only to specific other THESEUS routines; this makes a lot of icky error-checking necessary.

2 TYPES OF THESEUS ROUTINES

At this writing, there are approximately 100 routines in the package. This total includes approximately 90 routines that are ALGOLW-callable; these are described in [4]. The rest of the routines are "internal," i.e., they are not ALGOLW-callable, they are not known to the User, and they provide auxiliary services that are used by many of the other routines in the package. The number-conversion routines are examples of the "internal" routines.

In the rest of this document, the term "THESEUS routines" will be used to refer only to the User-known, ALGOLW-callable routines of the package. The "internal" routines will be referred to as such.

Excluding the internal routines, the package can be divided into seven mutually-exclusive groups of routines according to how each routine can be legally invoked by the User:

- The routines BUTTON, PICK, KEYBOARD, and LOCATOR can be invoked only as part of the expression passed to the POLL and WAIT routines. They are referred to as "poll routines."

- The routines VRP, VPN, VPD, PARALLEL, PERSP, VIEWUPnD[3], WINDOW, and VPORT can only appear as (part of) the last parameter to the ADDVIEW and CHGVIEW routines, i.e., as part of a viewing transformation parameter. They are referred to as "viewing transformation routines" or as "vtrans routines."

- The routines SCALEnD, TRANSnD, ROTnD, and TRANSMAT can only appear as (part of) the last parameter to the CALLOBJ and CHGCALL routines, i.e., as part of a modelling transformation parameter. They are referred to as "modelling transformation routines" or as "mtrans routines."

- The routines INTENS, PICKABLE, VECMODE, BLINK, CLIP, and INVERT[4] can only appear as (part of) an attribute parameter in CALLOBJ, CHGCALL, BEGSEG, CHGSEG, ADDVIEW, and CHGVIEW. They are referred to as "attribute routines."

- The EXTENTnD routines can only appear in an OPOBJnD call as the EXTENT parameter. They are called "extent routines"

- The SIZE routine is in a class by itself: it can only appear as the last (size) parameter in OPOBJnD and in BEGSEG.

- All other THESEUS routines can only be called directly, i.e., invoked in an ALGOLW procedure call statement rather than as a

---

[3]This construct appears throughout this document as an abbreviation for "VIEWUP2D and VIEWUP3D".
[4]And EXTCON and SIZECON, if they ever become implementable.

parameter (or part of one) to another THESEUS routine. They are referred to as "direct-call" or "direct" routines.

3 THE STARTQ13 GLOBAL DATA AREA

The ALGOL modu on BUGS is a runtime monitor for ALGOLW modu's. It loads and executes an ALGOLW mainline, but not until it has pointed register Q13 at a global data area (labelled STARTQ13) that includes constants (such as PI and MAXINTEGER), runtime variables (such as the output buffer pointer and stack pointers), and EQU's (describing the layout of such things as program and data segment headers). This area is created in the monitor by the AWXQ13 macro. If the parameter "DSECT" is specified in the macro invocation, the area is set up as a DSECT; since register Q13 is maintained as a pointer to the area by all ALGOL routines, all external runtime routines (e.g., input, output, record creator) can use such a DSECT form of the AWXQ13 macro to access this global information.

A modified version of the ALGOL modu, called THESEUS, must be used when running ALGOLW programs that invoke THESEUS routines. The modifications basically fall into two categories:

- those that make it possible for the User to run several ALGOLW mainlines with a single common graphics data structure[5].

[5]This provides a kind of basic (static) overlay facility -- e.g., the User could do initialization of a picture with one mainline and perform dynamics with a second one.

PROCRUSTES is initialized and terminated within the runtime monitor.

- Those that extend the STARTQ13 global area to contain global information needed by THESEUS routines.

These latter modifications basically appear in the THEQ13 macro. This macro works in exactly the same way as the AWXQ13 macro. In fact, it merely invokes the AWXQ13 macro, then adds on a couple of pages worth of global variables and EQU's.

The rest of this section will discuss the fields which appear in the THESEUS portion of the STARTQ13 area generated by the THEQ13 macro. It is assumed that the reader has a copy of the macro (or its expansion) handy: information provided in the internal comments will not be repeated here.

3.1 DIM, DIM2, DIM3, SD

The first several fields relate to some of the THESEUS command line options and are set by the runtime monitor. DIM can contain the value "2D", "3D", or "ALL" depending on whether the user specified the option "D=2D", "D=3D", or "D=BOTH" (respectively) on the command line. This is, of course, the dimensionality of the pictures the User will be allowed to create; the default is 2D. "BOTH" is stored

internally as "ALL" because that's the way PROCRUSTES likes to see it when PROCRUSTES is initialized.

DIM2 and DIM3 are constants against which comparisons may be made.

SD is the maximum stack depth allowed, i.e., the depth to which object calls may be nested in the graphic data structure; it defaults to 5.

3.2 BUFPTR, DIRPTR

These two fields are also set by the monitor and appear purely for convenience; they save various THESEUS routines from having to chase pointers in order to access the PROCRUSTES buffer header (BUFPTR) and the PROCRUSTES directory (DIRPTR).

3.3 OPOBJ, OBJDIMN, OBJDIMC

These three fields identify the currently open object. OPOBJ contains 0 if no object is open, X'8001' if the root object is open, and a positive integer (the object's name) if any other object is open. Note that the distinction between OPOBJ=0 and OPOBJ=X'8001' is a purely internal one: as far as the User is concerned, the root object is not directly

accessible (he uses the ADDVIEW, CHGVIEW and DLTVIEW routines) and, in fact, X'8001' (a negative integer) is not a valid name; if, for instance the User program invokes a THESEUS routine which requires an object to be open (e.g., BEGSEG), X'8001' will be treated as if it were 0. The only reason why the distinction is recorded at all is for efficiency in making calls to PROCRUSTES; e.g., if the User invokes ADDVIEW twice in a row while no object is open, there there is no reason for ADDVIEW to invoke PROCRUSTES twice to open the root object.

OBJDIMN and OBJDIMC record the dimensionality of the currently open object in integer and character form respectively -- each form is useful to different THESEUS routines. Neither field has any meaning if OPOBJ = 0 or OPOBJ = X'8001' (no User object is open).

3.4 THEFLAGS AND ASSOCIATED EQU'S

So why do we need these flags?

- HASXTENT: It is necessary to know if the currently open object has an extent because creation of a segment with an associated size (a BEGSEG call that contains a non-null SIZE parameter) is invalid if the object containing the segment has no extent.

- OPENSEG: It is necessary to know if a segment is open because certain routines (e.g., line and text primitives) are invalid otherwise.

- NEWOBJ: It is necessary to know if an object being opened by OPOBJnD is null in case the call to OPOBJnD specified a non-null SIZE or EXTENT parameter. In this case it is an error if a non-null object existed previous to this call; or if a non-null object is being re-opened, and does not have a size and/or extent to modify.

- HASATTR: Finally, it is necessary to know if the currently open segment has as attribute halfword associated with it (a BEGINSEG subblock) so that an ENDSEG subblock can be added when the segment is closed.

3.5 ATTRPTR, OPOBJPTR, CALLTPTR, VTRANPTR

These four fields are pointers to data that must be accessed by some of the THESEUS routines which are passed as parameters. Only one of these fields is needed at a time, so they are all EQU'd to the same halfword, but they are each given a different name for clarity. For example, when BEGSEG is called, it is passed a (possibly null) attribute routine; just before BEGSEG invokes this routine, it will fill in

ATTRPTR with the address of the attribute halfword to be diddled with.

3.6 CALLTYPE AND ASSOCIATED EQU'S

CALLTYPE can take on seven different values (the EQU's following it) corresponding to the seven different places THESEUS routines can be invoked from (see section 2). The first thing any THESEUS routine does upon being called is to make certain that it was called from a legal place. CALLTYPE is initialized to 0 ("direct-call") because the Wonderful World of THESEUS can only be entered by a call to a "direct-call" routine (all other routines are invoked as parameters of direct call routines). For instance, if BEGSEG is called it first makes certain that CALLTYPE=0 ("direct-call"); if so, before BEGSEG invokes the (possibly null) attribute routine passed to it, it sets CALLTYPE to 3 ("attribute call") so that the called parameter can be checked, and before BEGSEG returns to its caller (the User's ALGOLW program), it resets CALLTYPE to 0 ("direct-call").

3.7 DEFATTR

DEFATTR is an attribute halfword set to all the default values.

3.8 POLLINDX

POLLINDX is a variable used in conjunction with calls to POLL and WAIT. These routines expect to receive as a parameter a logical expression consisting of OR's of the THESEUS poll-routines (PICK, LOCATOR, BUTTON, KEYBOARD) and they return the ordinal number of the first poll-routine in the expression that returns "TRUE". (POLL returns N+1 if all N poll-routines return "FALSE"). This is accomplished by having POLL and WAIT set PCLLINDX to 0 before evaluating the parameter. Each poll-routine adds 1 to POLLINDX as it is invoked. The final value of POLLINDX is then returned by POLL or WAIT.

3.9 VIEWDIM AND ASSOCIATED EQU'S

VIEWDIM is the dimensionality of the view being added by a call to ADDVIEW or modified by a call to CHGVIEW. Legal values (for which EQU's are provided) are 0, 2, 3. This information is needed by the view-transformation routine

passed as a parameter to ADDVIEW and CHGVIEW so that appropriate error-checking can be done. Unfortunately, it is necessary that VIEWDIM be set to 0 if neither ADDVIEW nor CHGVIEW is executing, because the INVERT attribute routine can only be called legally as (part of) the attribute routine passed to an ADDVIEW or CHGVIEW of a 2D object, and INVERT uses VIEWDIM to error-check this (VIEWDIM=3 => not a 2D object, but VIEWDIM=0 => not even invoked from ADDVIEW or CHGVIEW).

3.10 CALLDIM AND ASSOCIATED EQU'S

CALLDIM is used by CALLOBJ and CHGCALL in a way analagous to that in which VIEWDIM is used by ADDVIEW and CHGVIEW (see section 3.9).

3.11 CALRBASE, ERRADDR

These two fields are used in conjunction with error-handling (see section 4). Error messsages are printed by the ALGOLW error-handler, AWXERROR. This routine needs, among other things, the address at which the error occurred (so that it can look up the coordinate number) and the address of the segment base for the program segment in which the error

occurred (so it can find the correct coordinate table). When an error is detected within a THESEUS routine, we want to print the coordinate of the ALGOLW statement which invoked THESEUS -- obviously, there are no coordinate numbers within THESEUS routines (they are assembler routines). Therefore, whenever THESEUS is invoked directly from the User's ALGOLW program, the base address of the caller's program segment and the return address are saved (in CALRBASE and ERRADDR, respectively) so that this information will be available in case AWXERROR is called.

3.12 CURENABL AND ASSOCIATED EQU'S

CURENABL is used to record the current status of the cursor. EQU's are provided to indicate that the cursor is not in use, that it is enabled as a locator, and that it is enabled as a pick.

3.13 SCREEN, WORLD

SCREEN and WORLD are both four-entry real arrays. They are used by the number conversion routines to convert from fixed-point fractions to floating point and vice versa. These conversions are required because the SIMALE Standard Graphics Package can only work on fixed-point fractions, whereas ALGOLW

can only work with fractions in floating-point form. Some THESEUS routines require screen coordinates while others require User's world coordinates as parameters. The first case merely requires converting floating point numbers in teh range [-1,1] to fixed point numbers in the same range. However, in the second case floating point numbers in an arbitrary range (symmetrical about 0) are mapped into fixed point numbers in the range [-1,1] (or vice versa). Hence two different arrays worth of magic numbers (SCREEN for conversions of screen coordinates, WORLD for world coordinates) are provided. Note that, until the User uses the RANGE routine, the world coordinate system defaults to [-1,1], so SCREEN = WORLD initially.

3.14 MINFIX, MAXFIX

MINFIX and MAXFIX are also used in number conversion. They remain constant and represent the maximum and minimum possible fixed-point fractions.

3.15 IDMATR2D, IDMATR3D

IDMATR2D and IDMATR3D are real arrays representing the identity matrix for 2D and 3D objects, respectively. They are used in initialization of matrices.

3.16 DDIMLIST, QDYNLIST, ADDMLIST, ADDSLIST, OPENLIST

These five lines are PROCRUSTES parameter lists, created with the list-form of the CALL macro (see [5]), for the five PROCRUSTES routines most commonly called by THESEUS routines. Calls to these PROCRUSTES routines are made with the E-form of the CALL macro, specifying the appropriate parameter list. Note that all of the parameters whose addresses appear in these lists also (of necessity) appear in the STARTQ13 area. The addresses in these parameter lists are should never be changed, with one exception: the PROCRUSTES ADDMARK routine treats its second parameter as optional, depending on whether of not its address is 0; therefore, the second entry of ADDMLIST should always be set (to 0 if the parameter is omitted, probably to A(ENAME) if it is present) before calling ADDMARK (or else the ADDMLIST parameter list should not be used).

3.17 PROCRUSTES PARAMETERS

The last 13 fields of the STARTQ13 area are the parameters pointed to by the PROCRUSTES parameter lists (see 3.16). These fields are variables. They are included in the STARTQ13 area for two reasons: (1) the AWXRHV macro which THESEUS routines use to access their parameters, can mung the registers, so parameters must be stored in memory anyway -- this is a convenient place, especially since most of those parameters get used in PROCRUSTES calls anyway; (2) this enables the PROCRUSTES parameter lists to appear in the STARTQ13 area also, saving space since they are each used byin many THESEUS routines.

4 ERROR HANDLING

AWXERROR parms

monitor changes

ENTERTHE

5 EXEC'S

There exist three EXEC's that may be useful to those maintaining THESEUS. All three are ridiculously simple and do a minimum of error checking.

5.1 GENTHE

This EXEC generates a new copy of the THESEUS runtime monitor. It takes no parameters. It assumes the existence of the file THESEUS TEXT. All it does is to GMSLINK this deck with the library PROCRUST TXTLIB (from which external references to PROSTART, PROTERM, and PROCRUST are resolved). A MODU map is produced. The symbols AWXSL006 (ALGOLW timer routine), AWXERROR (ALGOLW error handler), and PROCRUST (the PROCRUSTES global data area) are GLOBALed so that references to them by User MODU's will be resolved when the User MODU's are loaded on BUGS (under control of the monitor).

Note that AWXRCTBL will be unresolved after executing GENTHE -- this is o.k. AWXRCTBL is the record class table; it will appear in the User's MODU and be resolved at load time.

5.2 THEASM

This EXEC should be used to assemble any THESEUS routine, including internal routines and the runtime monitor. It takes the same parameters as ASMG. It merely does a BUGSGBL A of the necessary macro libraries (AWXALL, AWXM4A, PROCRUST, AWXRUNT), followed by a BUGSASM AL of the parameter list passed to THEASM.

5.3 THELINK

This EXEC is needed by both the Users and maintainers of THESEUS and should be made available through the COMMON segment. It is used to link a META4A ALGOLW text deck containing calls to THESEUS routines. Its use and format are identical to that of the ALGLINK EXEC described in [3], with two exceptions: (1) only seven files may be specified, (2) besides AWXSL006 and AWXERROR, the name PROCRUST may be unresolved -- it will be resolved at load time on BUGS.

6 LIBRARIES

All of the macro and text libraries described in this section must be available to anyone doing maintenance of THESEUS; THESEUS TXTLIB must also be available to the User for linking ALGOLW text decks containing calls to THESEUS routines. These libraries are accessed by the EXEC's described in section 5.

6.1 MACRO LIBRARIES

6.1.1 AWXALL, AWXM4A

These libraries contain the AWX-macros written by Chuck Sorgie to permit interfacing of assembler routines with ALGOLW routines. AWXALL contains those macros that perform correctly for both the /360 and BUGS, while AWXM4A contains those macros for which a special version had to be written for BUGS. See [4] for a description of these macros.

6.1.2 PROCRUST

This library contains the macros written by Russ Burns for use in the PROCRUSTES subroutine package. The most important macro in the library is CRUSTY, which generates a dsect describing the layout of the PROCRUSTES global data area, giving assembler routines (including those comprising THESEUS) access to the interactive devices, dials, PROCRUSTES buffer, etc. Other macros from this library which are used by THESEUS are:

- ISCAN, which generates a dsect describing the interrupt scanout area.
- DIRECTOR, which generates a dsect describing a PROCRUSTES directory entry.
- BHEADER, which generates a dsect describing the PROCRUSTES buffer header.

6.1.3 AWXRUNT

This library is used by those maintaining the ALGOLW runtime environment as well as those maintaining THESEUS. It contains three macros:

- QREGS generates the EQU's setting up the standard mnemonics for the ALGOLW Q-registers (see [4]), the META 4A registers, and the floating point registers. It is used in both ALGOLW runtime environment routines and THESEUS routines.

- AWXQ13 generates a dsect describing the ALGOLW runtime global data area, STARTQ13 (see section 3). It is used explicitly in the ALGOLW runtime environment routines; it is used implicitly (because it is invoked by the THEQ13 macro) in THESEUS routines.

- THEQ13 generates the same dsect as AWXQ13 (which it invokes) and then extends it with THESEUS runtime global data (see section 3). It is used only in THESEUS routines.

6.2 TEXT LIBRARIES

6.2.1 PROCRUST

This library contains the text decks for the PROCRUSTES subroutine package. The THESEUS maintainer uses this library to link the runtime monitor.

3.7 DEFATTR

DEFATTR is an attribute halfword set to all the default values.

3.8 POLLINDX

POLLINDX is a variable used in conjunction with calls to POLL and WAIT. These routines expect to receive as a parameter a logical expression consisting of OR's of the THESEUS poll-routines (PICK, LOCATOR, BUTTON, KEYBOARD) and they return the ordinal number of the first poll-routine in the expression that returns "TRUE". (POLL returns N+1 if all N poll-routines return "FALSE"). This is accomplished by having POLL and WAIT set PCLLINDX to 0 before evaluating the parameter. Each poll-routine adds 1 to POLLINDX as it is invoked. The final value of POLLINDX is then returned by POLL or WAIT.

3.9 VIEWDIM AND ASSOCIATED EQU'S

VIEWDIM is the dimensionality of the view being added by a call to ADDVIEW or modified by a call to CHGVIEW. Legal values (for which EQU's are provided) are 0, 2, 3. This information is needed by the view-transformation routine

resolved from this TXTLIB. THESEUS routines in the TXTLIB in turn make calls to (external) PROCRUSTES routines; however, the PROCRUST TXTLIB can not be used to resolve these calls -- if it were used, external references to the PROCRUST entry point would cause a copy of the global data area to be linked into the User MODU --this would mean that when the monitor loaded the User MODU, there would be two copies of the data area (one in the monitor, one in the User MODU) instead of one shared one.

We could just keep two different copies of PROCRUST TXTLIB around, a normal one (for linking the monitor) and one that excludes PROSTART/PROTERM/PROCRUST (to be used in conjunction with THESEUS TXTLIB for linking User MODU's). It was decided instead to incorporate the latter version into the THESEUS TXTLIB.

NOTE: This means that if changes are made to PROCRUSTES routines, updates must be made to both PROCRUST TXTLIB and THESEUS TXTLIB.

7 DYNAMOS

All dynamos are created with a byte offset of four. That is to say, each dynamo points to the beginning of the actual subblock data, skipping only the 4-byte subblock header.

The chart below indicates which THESEUS routines create and use dynamos.

dynamos created by:	are (legally) used by:
ADDVIEW	CHGVIEW
CALLOBJ	CHGCALL
LINEnD POINTnD POLYLINE POLYGON	CHGPTnD
TEXTnD	CHGTEXT
BEGSEG CRTDYNS	CHGSEG
CRTDYNE	CHGVIEW CHGCALL CHGPTnD CHGTEXT (depends on type of element for which CRTDYNE is actually called)

8 CALLERS OF PROCRUSTES ROUTINES

This section provides, for each PROCRUSTES routine called from THESEUS, a list of the THESEUS routines that call it.

CALLDDIM	QUERYDYN	ADDSUBLK	ADDMARK
ADDVIEW	CHGVIEW	ADDVIEW	ADDVIEW
CHGVIEW	CHGCALL	CALLOBJ	BEGSEG
DCLOBJnD	CHGSEG	BEGSEG	CHKDE
NULLOBJ	CHGPTnD	ENDSEG	CLOSELEM
DLTOBJ	CHGTEXT	EXTENTnD	
RNMOBJ	CRTDYNE	SIZE	
SWPNME	CRTDYNS	LINEnD	
CALLOBJ	CHKDE	POINTnD	
CHGCALL	CLOSELEM	POLYLINE	
CRTDYNS		POLYGON	
		TEXTnD	
		CHKDE	
		CLOSELEM	

OPENIMAG	ADDSEGMT	DELSEGMT	CALLRDIM
ADDVIEW	ADDVIEW	DELVIEW	OPOBJnD
OPOBJnD	BEGSEG	DLTSEG	DCLOBJnD
DCLOBJnD			

DELDYNAM	ADDDYNAM	DELIMAGE	BLKTYPE
DLTDYN	CRTDYNE	NULLOBJ	CRTDYNS
	CRTDYNS	DLTOBJ	

Also, PROSTART and PROTERM are called from the THESEUS runtime monitor.

9 CALLERS OF INTERNAL ROUTINES

This section provides, for each THESEUS internal routine, a list of those THESEUS routines that call it.

FIXS	FIXW	FIXSP	FIXWP
SIZE	CHGPT	PICKEXPL	LINEnD
INTENS	POLYLINE	KEYVIEW	POINTnD
PICKJOY	POLYGON		EXTENTnD
PICKTAB	FIXWP		TEXTnD
PROMPT	CHGCALL		CHGPTnD
POSPICK			
FIXSP			

CHKDE	CLOSELEM	FLOATS	FLOATW
CALLOBJ	CALLOBJ	READLOC	LOCATOR
LINEnD	LINEnD	READJOY	
POINTnD	POINTnD	READTAB	
POLYLINE	POLYLINE	READDIAL	
POLYGON	POLYGON		
TEXTnD	TEXTnD		
BEGSEG			
ADDVIEW			

TRIGROFF	KEYOFF	KEYON
DISPICK	TRIGROFF	BUTKEY
DISLOC	DISBUT	PICKJOY
		DISBUTS
		LOCJOYE

ENTERTHE is called from every THESEUS routine (except the internal routines).

10 REFERENCES

[1] META 4A Principles of Operation

[2] Harold Webber and Russell W. Burns, The SIMALE Standard Graphics Package Preliminary Description

[3] Howard Koslow, Programmer's Guide to ALGOL_W

[4] Ingrid Carlbom, THESEUS

[5] META 4A Assembler User's Guide