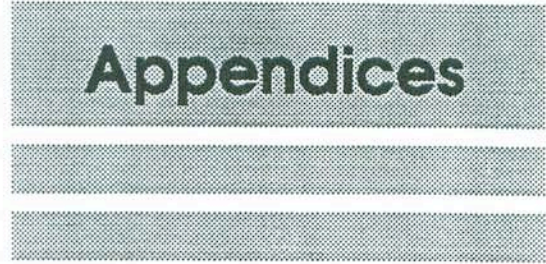


Appendices



Appendix A

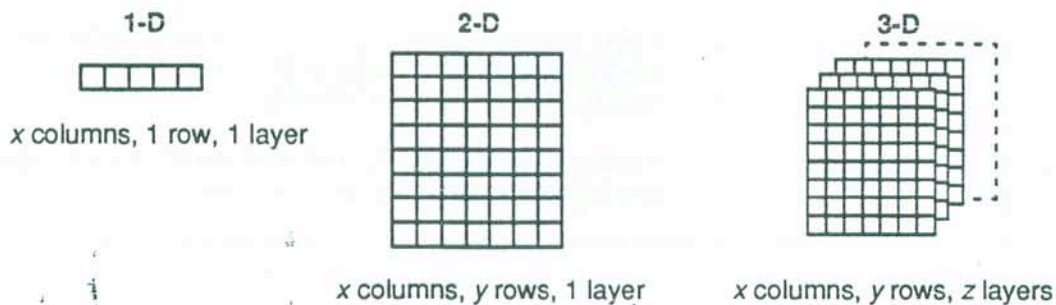
PICTURE TYPES

Overview

This appendix gives a summary of Semper picture types and explains some of the terms used in this manual. It details picture dimensions, class and form.

Picture dimensions

A Semper picture can have 1, 2 or 3 dimensions, as is illustrated in the following diagram:



Picture classes

Semper assigns a picture to a class according to its data type, as is shown in the following table:

No	Class	Description
1	<i>Image</i>	a sampled picture (most data fall into this class)
2	<i>Macro</i>	the text of a numbered macro
3	<i>Fourier</i>	the Fourier transform (<i>ft</i>) of an image
4	<i>Spectrum</i>	the intensity (modulus squared) of the <i>ft</i> of a picture
5	<i>Correlation</i>	the cross or auto-correlation function between pictures
6	<i>Undefined</i>	any data not otherwise classifiable
7	<i>Walsh</i>	the Walsh/Hadamard transform of a picture
8	<i>Plist</i>	a list of position coordinates with associated information
9	<i>Histogram</i>	the histogram (of the pixels) of a picture
10	<i>Lut</i>	a framestore look-up table

Semper 6 Command Reference

Picture Forms

Semper can store picture data in one of four different forms. Each form offers a different trade-off between storage and precision requirements. The following table describes the four types of picture storage. Note that the values of integer, fp and complex forms are, to some extent, machine dependent. The values given below may be considered as typical.

No	Form	Description
0	<i>Byte</i>	1 byte storage. 0 to 255, unsigned integers only. A compact but restrictive form of storage as negative values can easily arise in processing
1	<i>Integer</i>	2 byte storage. -32768 to 32767 or wider, integers only. <i>Integer</i> storage is less restrictive than <i>Byte</i> but still compact
2	<i>Fp</i>	4 byte storage usually. 1-35 to 1+35 or wider, either sign, integral or fractional (floating point). Typically one part in a million precision over the full range
3	<i>Complex</i>	8 byte storage usually. Pairs of <i>fp</i> values representing real and imaginary parts of a complex value

Further information

To see the dimension, class and form of a picture, for example picture number 2, type the following command:

```
examine 2
```

or type the command **examine all** for details of all pictures on a current device.

For further detail of Semper pictures, refer to the following manual:

Advanced Users' Guide

contained in the *Semper 6 Guide*.

Appendix B

SEMPER EXPRESSION

SYNTAX

Overview

This manual frequently describes commands that require a *valid Semper expression*. This appendix defines the syntax of a valid expression.

You may use an expression wherever Semper requires a simple numerical or logical value. The following are valid components of an expression:

- numbers
- variables
- function calls

All of the above supply a numerical value. Note that a *logical* value is represented by the number 1 or 0, where 1=*true* and 0=*false*. If Semper finds a numerical value where it expects a logical value it interprets a zero value as *false* and a non-zero value as *true*. (For further detail of variables and functions, refer to the *Advanced Users' Guide* contained in the *Semper 6 Guide*).

Expression Operators

There are three types of expression: **arithmetical**, **relational** and **logical**, in which the following operators can be used:

- **arithmetical** ^ * / + - :
- **relational** < > = ~= <= >=
- **logical** ~ & |

Note that ^ specifies exponentiation (raising a number to a power).

The colon operator : is a special operator that you use to combine device and picture numbers into a single numerical value. For example, 1:45 represents the numerical value 1045 and refers to picture 45 on device 1. If you do not specify a number before the colon operator, the current device number is assumed, (the value is held in the variable *cd*).

The logical operators ~ & | specify the logical operations NOT, AND and OR respectively.

Semper 6 Command Reference

Operator priority

The order in which Semper evaluates nested expressions depends on the relative priorities of the operators used in each expression (however you can control the order of evaluation using brackets). The order of priorities is listed here, with equal priority operators on the same line:

High Priority	:
	^
	* /
	+ -
	< > = ~= <= >=
	~
	&
Low Priority	

Functions

Function calls return a numerical value. Each call consists of a function name followed by one or more argument values enclosed in brackets (use commas to separate multiple arguments), for example:

exp(3) root(x) sin(theta) mod(3.6, -12.3)

The following functions are recognised by Semper:

sin(x)	sine (x in radians)
asin(x)	arcsin (result in radians), for mod(x) <= 1
cos(x)	cosine (x in radians)
acos(x)	arccosine (result in radians), for mod(x) <= 1
tan(x)	tangent (x in radians)
phase(x)	arctan (result in radians)
phase(x,y)	arg(x+iy) (result in radians), for mod(x) <= 1
exp(x)	exponent (e to the power of x)
ln(x)	ln (natural or Napierian log), for x>0
min(x,y)	lower of values x,y
max(x,y)	higher of values x,y
mod(x)	modulus (absolute value)
mod(x,y)	modulus(x+iy), mod(x,y) = root(x*x + y*y)
msq(..)	= mod(..) squared
root(x)	square root, with x>= 0
rem(x,y)	remainder when x is divided by y, rem(x,y) = x - fix(x/y)*y
round(x)	nearest integer to x
fix(x)	next integer towards zero from x
p(x,y,z)	value of pixel x,y,z of current picture
re(x,y,z)	real part of pixel x,y,z, same as p(x,y,z)
im(x,y,z)	imaginary part of pixel

Semper 6 Command Reference

<code>and(n,m)</code>	bitwise and of <code>round(x)</code> and <code>round(y)</code>
<code>or(n,m)</code>	bitwise or of <code>round(x)</code> and <code>round(y)</code>
<code>not(n)</code>	bitwise not of <code>round(x)</code>
<code>rad(x)</code>	angle in radians equivalent to <code>x</code> degrees
<code>deg(x)</code>	angle in degrees equivalent to <code>x</code> radians
<code>ifelse(x,y,z)</code>	returns <code>y</code> if <code>x</code> is non-zero and <code>z</code> if <code>x</code> is zero
<code>set(variable name)</code>	returns 1 if named variable is set; otherwise 0

Note that Semper does not confuse function calls with variables of the same name. For example, `min(x,y)` and `min` may be used quite independently.

Examples

Expressions provide a powerful and general way of combining and manipulating numerical data. A number of examples is given below to illustrate the range of possibilities:

<code>56</code>	<code>-1.5</code>	<code>x</code>	<code>192*64</code>
<code>n+3</code>	<code>.5-x</code>	<code>min*2.4</code>	<code>mod(x,y)5</code>
<code>(a+b)/2</code>	<code>scale*(r+.2)</code>	<code>1/(x*x+a*a)</code>	<code>exp(-msq(x,y)/50)</code>
<code>(x-fix(x/y)*y)</code>	<code>deg(pi)</code>	<code>sin(rad(60))</code>	<code>-set(erase)</code>
<code>x=y<=5</code>	<code>a<1 a>10</code>	<code>and(or(not(a),not(b),or(a,b)))</code>	

Appendix C

SEMPER KEYS

AND OPTIONS

Overview

This appendix provides a summary of the following keys and options that can be used with Semper commands:

- *general* keys and options
- keys for display marking
- commonly used options
- subregion keys and options

General keys and options

The following keys and options are called *general* and are usually not listed under the syntax of a command. They can be used with any command (although they may be ignored if the context is inappropriate).

byte, integer, fp, complex	These options determine the form of a command output. By default, Semper commands produce output in the same format as the source picture, unless you override this by using one of these options
erase	The erase option erases display partitions before pictures are displayed in them. The default is noerase .
view	You can use the view option with any command that displays pictures to make a display picture visible. For example, you can use view with the mark command if the display that you want to mark is not initially visible, or with commands such as erase and ramps .
re, im	These options limit the annotation on a complex display to the real or imaginary part. By default, Semper annotates the real part and repeats the annotation on the imaginary part.

Semper 6 Command Reference

Mark keys

The following keys are used to mark a display. They are *general* keys but are listed under the syntax of a command in this manual, if a command uses display marking:

mark

This key may take the values *yes* or *no* (1 or 0):

If *yes*, the current display picture is marked.

If *no*, display marking does not take place.

If you do not specify the key, **no** is assumed. You can also specify any valid display picture number with the **mark** key, in which case the specified display picture is marked. You can turn off display marking permanently by typing **mark=no**, and permanently turn on marking by typing **mark=yes**.

mkmode

The **mkmode** key determines the style of annotation when marking points. You can specify the following values:

- 1 upright cross (default)
- 2 diagonal cross
- 3 upright box
- 4 diagonal box
- 5 single pixel

The default is **mkmode 1**.



mkmode 1



mkmode 2



mkmode 3



mkmode 4



mkmode 5

mksize

This key determines the size (radius) of the annotation. The total width of the mark is $2 * mksize + 1$. Note that **mksize** is ignored if **mkmode** is set to 5.

Semper 6 Command Reference

Other widely used options

The following options are commonly used and appear in the syntax description of many Semper commands:

preset	The preset option causes a command to use the current values of the variables <i>min</i> and <i>max</i> , instead of the actual pixel range, for scaling purposes. The default is nopreset .
verify	This option gives information about the action and results of a command at the console. The default depends on the particular command.
letter	This options adds lettering to a display that gives size, title and grey scale information for a picture. The default is to add lettering.
border	This option marks a border that outlines a display picture. The default is to mark a border.

Subregion keys and options

You can specify a subregion using the following keys and options:

- **size**
- **position**
- **left/right, top/bottom, near/far**
- **layer**

Use the **size** key to specify the dimensions of a subregion. **size** has three components which correspond to the x, y and z dimensions of the subregion.

If you do not specify a value for **size**, it defaults to the dimensions of the source picture. If you only specify one dimension, the y dimension defaults to the x dimension and the z dimension defaults to the source picture z dimension. For example:

```
size 300, 200
```

describes a subregion 300 pixels across by 200 down, and:

```
size 250
```

defines a subregion that is 250 pixels square.

Use the **position** key and the **left/right, top/bottom, near/far** options to specify the position of a subregion. By default, Semper positions a subregion so that its centre coincides with the source picture origin. Use the **left** or **right** option so that it is placed with its left or right side coinciding with

Semper 6 Command Reference

the left or right side of the source picture. The options **top/bottom** and **near/far** have a similar effect on the other two directions. For example:

```
size 100 bottom left
```

defines a 100 pixel square subregion at the bottom left of the source picture. The **position** key allows you to specify an offset for the subregion. The centre of a subregion is positioned exactly at the *x, y* position you specify if you do not use any of the subregion options, for example:

```
size 100 position 200, 350
```

defines a 100 square subregion centred on the position 200, 350.

Multi-layer subregions

You can specify a layer for a subregion using the **layer** key. For example:

```
size 300, 200 layer 5
```

defines the front layer of a picture with 5 layers 300 by 200 in size. You can also specify a range of layers, for example:

```
size 300, 200 layers 2, 4
```

specifies a three layer subregion, starting at layer 2.

Appendix D

PARTICLE

PARAMETERS

Overview

The **analyse** command records 25 different parameters for each particle it finds in a particle parameter list (*ppi*), held by Semper as a class *Plist* picture. Each parameter has a name and a variable associated with it. The *name* is used to specify or select individual or sets of parameters in later printout, sorting or display commands. The *variable* is used to store individual parameter values for further manipulation.

The table given below details the parameters recorded by **analyse**. An explanation of the individual parameters is given after the table.

Name	Variable	Parameter description
xref, yref	<i>xr, yr</i>	reference point
id	<i>pid</i>	particle identifier
parent	<i>pa</i>	parent identifier
holes	<i>h</i>	number of holes
background	<i>bg</i>	background flag
contact	<i>ec</i>	edge contact flag
xmin, xmax	<i>x1, x2</i>	limits – min, max x
ymin, ymax	<i>y1, y2</i>	limits – min, max y
hferet, vferet	<i>hf, vf</i>	horizontal and vertical feret diameters
aferet, bferet	<i>af, bf</i>	feret diameters – 45 degrees, 135 degrees
hproj, vproj	<i>hp, vp</i>	horizontal and vertical projections
perimeter	<i>p</i>	perimeter
area	<i>a</i>	area
xcen, ycen	<i>xc, yc</i>	centre of area
mmin, mmax	<i>m1, m2</i>	principal second moments of area – min, max
angle	<i>theta</i>	orientation
circularity	<i>c</i>	circularity

Semper 6 Command Reference

Reference point

The *reference point* gives the position of the right-most pixel of a particle at its bottom-most row and is therefore guaranteed to lie on the edge of a particle.

Particle identifier

The *particle identifier* provides a particle with a unique identifying number. Identifiers are assigned from 1 onwards in the order in which the particles are found during a picture scan.

Parent identifier

A *parent identifier* gives an identifier for the region surrounding a particle. Note that this parameter is to be used by later releases of Semper. Currently all parent identifiers are set to zero.

Number of holes

The *number of holes* means the number of background areas that are enclosed within a particle.

Background flag

A *background flag* indicates that the 'particle' is a background area (hole) enclosed inside another particle. This is to be used by later releases of Semper. Currently all background flags are set to zero.

Edge contact flag

The *edge contact flag* of a particle is set to 1 if the particle makes contact with an edge of an analysed picture, and to zero otherwise.

Limits

The *limits* of a particle are the minimum and maximum x and y values of all pixels belonging to a particle.

Feret diameters

The *feret diameters* of a particle are feret or *caliper* diameters in four different directions – horizontally, vertically, at 45 degrees anti-clockwise from the positive x axis and at 45 degrees clockwise from it. The diameters provide rough size indicators and can be used to detect elongation (if there is a large difference between minimum and maximum diameters) and orientation (using the largest diameter).

Projections

The *horizontal* and *vertical projections* of a particle record the total number of right-facing and upward facing edge pixels for an object, that is, the total projected edge length facing the relevant direction.

Perimeter

The *perimeter* is the distance around the outside of a particle.

Semper 6 Command Reference

Area

The *area* of a particle is simply the number of pixels in a particle.

Centre of area

The *centre of area* parameters provide the mean *x* and *y* values for all pixels belonging to a particle. In many cases this may be more convenient to use than the particle reference point, as the centre of area lies roughly at the centre of a particle. Note that it may not actually lie inside a particle if the particle is hollow or convoluted.

Principal second moments of area

A *second moment* is the mean square distance of all pixels about a line through the centre of area of a particle. The *principal second moments* recorded by **analyse** are the second moments with respect to a pair of mutually perpendicular axes in directions that achieve maximum and minimum moments. Note that their square roots provide a reasonable measure of a particle's mean external dimensions (actually a factor of 3 to 4 times the square root, depending on the shape of the particle), and their ratio a measure of elongation.

Orientation

The *orientation* of a particle is the angle in radians clockwise from the positive *x* axis to the axis giving the lowest second moment of area, that is, the *long* axis of a particle.

Circularity

The circularity of a particle measures, on a scale from 0 to 1, how circular a particle is. It uses the following equation:

$$4 * \pi * \text{area} / (\text{perimeter}^2)$$

Appendix E

ERROR

MESSAGES

Overview

This appendix lists in numeric order the error messages given by Semper 6. It explains each error message and suggests possible causes.

Error messages

1 *I/O error on device number at file number block number*

Possible causes of error 1 include:

- inability to read/write a damaged or dirty tape
- hardware failure in tape drive/controller
- slight alignment differences between tape drives used to read and write tape

There isn't much you can do about this, as the offending block will already have been tried several times. You can set the variable *mtpass=yes*, which causes Semper to ignore the error while reading, and so allows you to continue processing with some data incorrectly transferred.

2 *Unexpected EOF on device device number*

Error 2 means that an end-of-tape mark has been reached unexpectedly on tape. Likely causes include:

- a malformed tape
- a hardware failure
- mistakes in a new Fortran routine of your own writing

3 *Bad value for keyword/variable*

Error 3 means that the named variable or keyword has an inappropriate value, for example, a negative radius for a circle. It may be caused indirectly by omitting a vital key for which there is no useful default, as this is usually equivalent to giving it a zero value. Note that only the first three letters of the keyword/variable appears in the message.

4 *Abandoned*

Error 4 occurs when Semper acknowledges an abandon request from the terminal.

Semper 6 Command Reference

5 *Bad size for picture picture number*

Possible causes of error 5 include:

- zero or negative picture dimensions requested
- picture row length too great for internal row buffers (**show system** lists maximum lengths for each form)
- non-factorisable dimensions in operation such as **rotate** (that is, dimensions not factorisable into factors 2,3,4 and 5 with at least one factor 4. The command **show sizes** lists the acceptable sizes)
- dimensions not a power of two in an operation involving *Fourier* or *Walsh* transformation
- inappropriate picture dimensions, for example, a 2-D picture where 1-D is required.

6 *Bad class for picture picture number*

Error 6 means that you have tried to apply an operation to data to which it is not appropriate. Typical examples include:

- *Fourier* transforming a picture which is already a transform
- calling an *Image* picture as a macro (e.g. @56)
- displaying a macro, or doing almost anything except **listing** or **editing** it
- histogram equalising with an *Image* rather than a *Histogram*
- loading a non-*Lut* picture as a look-up-table
- using another class where a *Plist* is needed, for example, as a mask boundary

Probably you have typed a wrong picture number; otherwise, you may have misunderstood how the command you are using works, and should ask **help** about it again (perhaps using **help/full...**).

7 *Undefined label: label*

Error 7 means that you (or a macro or program you are using) have tried to jump to the label indicated, but that the label cannot be found. The error context printout will tell you whether the problem is inside a program. Likely causes include:

- a mistyped name in the label or **jump** command
- trying to jump to a label outside the program
- trying to jump into a **for** loop
- trying to jump to a previous or subsequent line interactively

8 *I/O error on device number at block number ff*

Likely causes for error 8 include:

- inability to read/write information on a damaged/dirty disc
- hardware failure in a disc drive/controller
- writing to a file to which you have read access only (indicates a problem in the primitive routine *MCDC61*)

Little useful response is possible. No data loss will occur if the error arises during reading, but its extent is not usually very serious in any case, normally resulting in directories or picture data being backed-up to their state shortly before the error.

Semper 6 Command Reference

9 *Outside picture*

Likely causes for error 9 include:

- using or setting a pixel with coordinates that are out of range
- using a bad layer number
- using a subregion that does not overlap the picture at all
- setting a picture coordinate origin outside the picture
- mistakes in a new Fortran routine of your own writing

10 *Message depends on error condition*

Error 10 is not in fact reported by Semper. Commands that produce very specialised error messages of their own follow these by returning error 10 to the interpreter, which responds by returning to the terminal for fresh instructions as usual, but without printing any further message.

11 *Disc full – can't open picture number, on device, device number*

Error 11 means that there is insufficient free space left on a disc device for a new picture that you are creating. The things you might try in response are:

- deleting unwanted pictures from the device in question
- using a more compact form for your output picture (e.g. **lmean 50 byte**)
- storing other pictures in more compact forms (e.g. **copy 41 byte**)
- creating a new disc device, temporarily if necessary, with sufficient space (for example **assign new name 'newname' size ..**)

It is *not* worth trying to **compress** the device; the error message means that there is insufficient space in total, not simply that it is fragmented.

12 *No structure in picture picture number*

Error 12 means that some kind of picture scaling operation is impossible, because the picture minimum and maximum values are equal. Likely causes include:

- displaying a picture (or subregion) in which all pixels are constant
- setting *min*, *max* equal when specifying **preset** (e.g. **scale preset range 1, 10** when *min* and *max* are both zero)
- using correlation functions to register two images one or both of which has all pixels equal

13 *Can't set variable*

Error 13 has two likely causes:

- you have tried to reset a protected variable such as *pi* (**help variables** tells you about these)
- you have tried to set too many variables simultaneously (**show system** tells you the maximum number supported by your installation). Use **unset** to clear those you don't really need

Semper 6 Command Reference

14 *Unknown command: command*

Error 14 means that your command (after any prefixed labels, conditionals and assignments have been stripped off) begins with a name that is not a currently defined command. Probably you mistyped it; if in doubt, try **show commands**, which lists all commands recognised by your installation.

15 *Display device is write-only*

Error 15 means that you have tried to recover picture data from the display device in an installation where your display hardware is not capable of reading back such data. Note that trying to process a display picture *in situ* (e.g. **sharpen display**) in fact involves this kind of read-back, as the picture data must be recovered by your computer before the sharpened version can be calculated.

16 *Unknown function: function name*

Error 16 means that you have used an unrecognised function name in an expression, for example, **type six(theta)** instead of **type sin(theta)**. It can also be caused by mistyping: for example, **extract angle t(2** instead of **extract angle t*2** tries to treat *t* as a function name because of the bracket following it. **help functions** lists all functions recognised by Semper.

17 *Bad syntax*

Error 17 means that Semper cannot understand a command; possible reasons include:

- mistyping commands so that they are unrecognisable, e.g. **p x 2** or **[x=2** for **p x=2**
- omitting essential items from commands, e.g. **jump** without a label name
- including spurious additional items, e.g. **extract angle=.3** (which should have no = sign)
- typing **name#** without a following subscript variable

18 *Expression stack overflow*

Error 18 means that you have used an expression that is too complicated for Semper. It is likely to occur in **calculate** commands (for which each intermediate result involves a whole picture row). To cure the problem, simply split your expression into two simpler ones.

19 *Can't unset variable*

Error 19 is reported if you try to **unset** one of the protected or fixed variables such as *pi* or *cd* (type **help variables** for a list).

20 *Bad syntax in expression*

Error 20 means that an expression you have used is wrongly constructed, e.g. **cos()**, **s+*6** or **root(2,3)**. **help expressions** explains the syntax of expressions if you are in doubt.

Semper 6 Command Reference

21 *Command line too long*

A likely cause of error 21 is that a macro expression has caused a command line to exceed the permitted length. Type **show system** to see the maximum length of a command line for your installation. Note that numbered macros are an old feature that will be removed in a later release. Converting the macro into a library program using the command **add** should remove the problem.

22 *Bad numerical operation*

Likely causes for error 22 include:

- division by zero, e.g. **type 1/x** when *x* is zero, or **calculate :sel/:2** when some pixels of picture 2 are zero
- taking the **ln** or square **root** of a negative number, e.g. **mark radius root(t)** when *t* is -2

23 *Bad position on device device number*

Error 23 means that you have referred to an inaccessible position on the tape device indicated, such as a file beyond the current end of tape, a negative file number, or a block number larger than the number of blocks in a file.

24 *RUN file already active*

Error 24 means that you have nested a **run** call from within a run file. If you want to call other programs from a run file load them into a program library file and call them using the **library** command.

25 *Unset variable: variable*

Likely causes of error 25 include:

- using an unset variable as a key value or in an expression, e.g. **if x>t** when *t* is unset
- providing no value for a key which you are required to set because no useful default can be provided
- neglecting to set variables needed by a command, e.g. **spc levels h3 list** when *h3* is unset
- syntax errors, e.g. **ask 'value', x** for **ask 'value' x** (in the former, *x* is treated as part of the prompt string)

26 *Involves backward tape movement*

Error 26 means that you have tried to access a tape file before the current tape position, in an installation which cannot backspace or rewind tapes.

Semper 6 Command Reference

27 *Output to device number not allowed*

Likely causes for error 27 include:

- trying to output data to tape with any command other than **copy**, e.g. **lmean 50 to 3:1** when device 3 is a tape
- trying to alter existing data on a tape, e.g. **lmean 3:2** or **select 3:2; p0=1** or **title 3:2**

28 *Bad device or picture number: number*

Likely causes for error 28 include:

- using a zero or negative picture number, e.g. **3:0**, when not **copying** to tape
- using a picture number greater than 999 for a disc or tape
- using a display picture number greater than the maximum for your installation
- using a zero or negative device number
- using a device number greater than the maximum for your installation

29 *Bad medium for device device number*

Error 29 means that you have tried to do something to a device or picture that is inappropriate to its storage medium (disc, tape, display, help library). Likely causes include:

- **rewinding** a disc device
- **renumbering** a tape or display picture
- **compressing** a tape or display device, or asking for **directory** information about one
- doing anything to a help library other than **assigning** and **deassigning** it
- **marking** subregions on a disc or tape picture

30 *Picture picture number does not exist*

Most commonly, error 30 means that you have referred to a disc or tape picture that does not exist. Try **examine all** or **examine device..** to check what does exist, or **show device** to see how many pictures are on a tape device. Note that a display partition may exist without having a picture in it, so **show partition dis:3** can report details of the partition even when **examine dis:3** reports error 30.

31 *Bad tab position*

Error 31 means that a tab setting used in a **type** or **ask** command, or in the value for a textual key, requests a character position that is negative or too far right, e.g. **type 'values',#h,x,#v,y** when *h* is zero or *v* is 1e5.

Error number 32 is currently not in use

33 *Display display number is undersampled*

Error 33 means that you have attempted to recover a picture from, or process *in situ*, a display picture which has been undersampled to make it fit its partition. The picture therefore no longer has all the pixels it ought to have.

Semper 6 Command Reference

34 *Device device number is not assigned*

Error 34 means that you have tried to do something to a device number which is not currently assigned, and does not therefore refer to any particular disc, tape or display. **show devices** confirms the current list of assigned devices. Probably you have mistyped a picture number.

35 *Device device number does not contain programs*

Error 35 means that you have used a command that expects to find programs on a specified device, for example, **list all device 3**, when device 3 is a picture disc or help library.

36 *Bad FOR loop*

Likely causes for error 36 include:

- syntax errors in a **for** command, e.g. **for n=1**
- too many **for** loops active at once, e.g. **for..; for..; for..; for..; for..**
- bad loop increment values, e.g. **for n 1,2,0**

37 *Disc directory full - can't open picture number*

Error 37 means that insufficient space remains in the directory of a disc device for the recording of a new picture you are creating. It is likely only when the device is a **save** file, or when the directory has been deliberately created with a small number of slots to save space. But it often occurs if you have very large numbers of pictures within a device. You can ask **directory** about current directory slot usage. **compressing** the device may help, as each empty segment of disc requires a directory slot to record it.

38 *Insufficient display frames available*

Error 38 means that you are trying to display a multi-layer picture, probably full colour, for which each layer is output to a separate (successively numbered) display frame, and have run out of hardware display frames. If your installation provides four frames, for example, you can display full colour pictures in partition frames 1 to 3 and 2 to 4, but not in partitions which start with frames 3 or 4.

39 *Incompatible re-use of picture picture number*

Error 39 means that you are trying to process a display picture *in situ* in a way that requires two display pictures of different sizes to co-exist during the processing, for example, **extract display size 400** when the display is 300 square. (Differences in form do not cause problems, except that *Complex* pictures, for which real and imaginary parts are displayed side by side, are incompatible with non-*Complex* pictures).

Semper 6 Command Reference

40 *Display error*

Likely causes for error 40 include:

- hardware failure in the display device or interface
- using false or full colour look-up-tables in installations where the hardware does not support them
- using zoom factors or pan settings not supported by your display hardware

41 *Write-protected device or picture number*

Likely causes for error 41 include:

- altering a protected picture (processing it *in situ*, e.g. **lmean 51**, using the **p** command to change pixels, changing the title etc.)
- using an output picture that already exists and is protected, e.g. **lmean 50 to 51** when 51 is protected
- deleting a protected picture
- altering anything at all on a protected device

If in doubt about whether a picture is protected, ask **examine**. For devices, ask **show device** instead. Picture level protection can be removed or imposed via the **wp** command; device level protection is established when the device is **assigned** and cannot be altered – deassign and then reassign the device.

42 *Disc fragmented – can't open picture number, on device, device number*

Picture files require contiguous free space on a disc. Use the **compress** command to compact the free space. Type **directory** for a list of the available free space on your current device.

43 *Bad form for picture picture number*

Error 43 means that you are trying to use or store data whose form (byte, integer, fp or complex) is unsuitable in some way, for example, applying an operation only meaningful for complex data to a non-*Complex* picture.

44 *Assign/deassign failure on device device number*

Error 44 is reported for any kind of problem arising during **assigning** or **deassigning** devices; common causes include:

- mistyping the name of a disc file you are **assigning**
- **assigning** without option **wp** a disc file you are not authorised to alter
- deleting (via **deassign delete**) a disc file you are not authorised to delete
- **assigning** a **new** disc file for which space cannot be found
- **assigning** a tape or display device which is being used by someone else

Semper 6 Command Reference

45 *Device table full*

Error 45 means that you have tried to assign too many devices at once. **show system** tells you the maximum number permitted in your installation, and **show devices** lists the devices you have assigned already. You need to **deassign** some other device before trying again. Note that the **save** command assigns a further device temporarily (for the save file), and so can also give rise to this error.

46 *Picture picture number already exists*

Error 46 means that you have tried to **renumber** a picture to a new number that is not in fact currently unused.

47 *Row length overflow in picture picture number*

Error 47 usually means that Semper's internal row buffers are not large enough to process rows of the picture indicated in the data form that your command uses internally. This form is not under your control, and even when your source and output pictures are in byte form, the routine may in fact process pixels in *fp* form, and rows that fit the buffers in *byte* form may not do so in *fp* form. The only real solutions are breaking pictures up into smaller sub-pictures, or recompiling the entire system with longer row buffers.

48 *Bad type for display display number*

Error 48 means that the display indicated is of a type (2-D, 1-D, histogram, etc.) inappropriate for your command. Likely causes include:

- using the cursor in modes only possible for 2-D pictures (e.g. defining a closed curve) on a 1-D graph or histogram
- using a 1-D graph or histogram as source to a command, e.g. **copy dis:2 to 23** (the data cannot be recovered from these display types)

49 *Unknown macro: macro name*

Likely causes for error 49 include:

- mistyping the name of a macro, e.g. **@refion** for **@region**
- using the **@** character accidentally

50 *Bad block number: device number, block number*

Error 50 does not usually happen in Semper, and is likely to indicate a mistake in Fortran code of your own writing. Specifically, it indicates that the low level routine DISC has been asked to read or write an out-of-range block number on the device indicated.

Semper 6 Command Reference

51 *Unable to open any disc workspace*

Error 51 means that a command needing temporary work space on disc has been unable to find any or enough. Likely causes include:

- no disc devices assigned at all (**assign**)
- no disc device with sufficient contiguous space available (**compress**)
- all disc devices are write-protected (**wp**)

Only a few commands (amongst them **lmean**, **lvariance**, **lsd**, **sharpen** and **hp**) are in fact dependent on such work space.

52 *Picture picture number has a malformed label*

Error 52 does not usually happen in Semper, but may occur following a disc hardware fault. It indicates that the label of the indicated picture, which contains all the crucial information about the picture (dimensions, form, etc.) is corrupt. The only useful response is to delete the picture using the command **delete .. malformed**.

53 *Display partition partition number does not exist*

Error 53 means that you are using the indicated display picture or partition number without defining where the partition is stored and how large it is. If you have not simply mistyped the number, use the **partition** command before continuing.

54 *Magnified region too large*

Error 54 can occur when you are magnifying a subregion with **display**. It means that you have magnified it so many times that it does not fit the display partition. Try again with a reduced value for **times**.

55 *Too many histogram channels*

Error 55 means that a histogram you are creating has too many channels for the counts to fit Semper's internal buffers. Try again with a smaller value for the **channels** key.

56 *Too many pictures open simultaneously*

Error 56 means that a command uses too many pictures at once. It may occur in a **calculate** command, e.g. **calculate :1+:2=:3+:4+:5+:6+:7+:8..**, in which case you must split the command into two simpler commands. Otherwise it may arise in new commands of your own writing. **show system** tells you the maximum number that may be open at once in your installation.

57 *Illegal item: item*

Error 57 means that your command is wrongly constructed in some way; likely causes include:

- you have used an illegal (possibly mistyped) name for a key or option, e.g. **extract anfle .3**
- you have omitted a required key name

Semper 6 Command Reference

58 *No pictures found*

- Error 58 means that you have applied a command to a group of pictures (e.g. **examine 20,40** or **copy 1,10 ..** or **save 3:20,3:99**) when no pictures exist in the indicated range.

59 *Output must not overwrite source*

Error 59 means that the output from your command would be stored in the same place as the source, and that the processing operation being carried out would fail in such circumstances. Semper only reallocates the same storage if there is no change of picture number, dimensions and form, so you cure the problem by directing output somewhere else (for example, **extract 50 to 51** rather than simply **extract 50**).

60 *Conflicting keys and options: key/option and key/option*

Error 60 means that the two keys or options indicated are mutually incompatible, invoking strictly alternative modes – e.g. **ctf add** adds to the source, and **ctf multiply** multiplies by it, but **ctf add multiply** is faulted.

61 *Bad layer for picture picture number*

Error 61 does not normally occur in Semper, but may occur in new routines of your own writing if you accidentally refer to a picture layer that does not exist.

62 *Command does not process multi-layer pictures*

Error 62 means that the named command currently makes no provision for multi-layer pictures in any simple way (such as repeating the operation for each layer, or effecting a 3-D generalisation of the 2-D operation). This error may be expected to occur more rarely in later releases of Semper.

63 *Bad origin for picture picture number*

Likely causes for error 63 include:

- setting a picture coordinate origin outside the picture bounds (in new routines of your own writing)
- using a *Fourier* picture whose origin is neither at the centre nor at the centre of the left hand column

64 *Bad key/option: key/option*

Error 64 means that you have used an option inappropriate to the particular use of the command, e.g. **lut 2 red ..** when **lut 2** is monochrome.

65 *All pixels zero*

A possible cause of error 65 is using a **fir** filter kernel (**fir with..**) with no non-zero values.

Semper 6 Command Reference

66 *Bad kernel size*

Likely causes for error 66 include:

- using a **fir** filter kernel with too many points in each row (the limit is installation dependent, but is at least 21)
- using a **fir** filter kernel with too many points in total (the limit is the maximum number of fp values that fit a row buffer. See **show system**)
- offering a 2-D **fir** filter kernel for **separable** application

67 *Picture picture number does not define a closed curve*

Error 67 means that the *Plist* picture indicated should define a closed curve and does not. Perhaps you have mistyped the picture number; for example, **mask with ..** requires a closed curve to define the boundary of the region inside/outside which the picture is to be masked. See **xwires.curve** for how to create such *Plists* if necessary.

68 *Bad look-up table number: number*

Error 68 indicates that you have used an out-of-range look-up-table number; negative, zero or greater than the maximum permitted for your installation (see **show system** for this limit).

69 *Look-up table number does not exist*

Likely causes for error 69 include:

- viewing with a lut you have not created (use **lut** to create luts)
- using **lut** to recover or alter a lut you have not created

show luts lists the currently defined luts.

70 *Bad display partition number: partition number*

Error 70 means that you have used a display partition number that is illegal in some way. Likely causes include:

- using a negative or zero partition number
- omitting the display device number (e.g. typing an ambiguous 4 for **dis:4**)

show partitions lists the partitions currently defined.

71 *Device device number is not a display device*

Error 71 means that you have tried to do something intended only for display devices (for example: **erase**) to the disc or tape indicated.

72 *Bad type for Plist picture number*

Likely causes for error 72 include:

- using the list *Plist* indicated when a curve is needed, e.g. **mark with ..**
- using the curve *Plist* indicated when a list is needed, e.g. **extract with ..**

Semper 6 Command Reference

73 *Bad device: partition number: number*

Likely causes for error 73 include:

- using a non-display device number when a display is needed, e.g. **mark 2:3** when 2 is a disc device
- using an out-of-range partition number (negative, zero or beyond the maximum your installation permits. See **show system**).

Error number 74 is currently not in use

75 *Bad display frame number: number*

Error 75 is caused by using an out-of-range frame number (negative, zero or beyond the maximum your installation permits). See **show devices**.

76 *Bad device number: number*

Error 76 means that the indicated device number is illegal (negative, zero or beyond the maximum permitted for your installation). See **show system**.

77 *Error message*

Commands that produce very specialised error messages of their own return error 77 and an error message which differs according to the context. You can see this message by using the **report error** command.

78 *3 points defining arc or circle are collinear*

Error 78 can arise in **xwires circle** or **xwires arc**, which seek to define a circle or a circular arc, if you mark three points in a straight line, as these would require an infinite radius.

79 *U and V not Independent*

Error 79 means that the vectors $(u, u2)$ and $(v, v2)$, used by your command, are parallel and must not be so, for example, because they are supposed to be base vectors of a 2-D lattice.

80 *Maximum for number of input points has been exceeded*

Error 80 arises in **xwires** if so many points are marked (**list** mode) or generated (**sampling** key specified in **curve** or **graph** mode) that internal workspace tables overflow. The only useful response is to repeat the operation with a smaller number of points.

Semper 6 Command Reference

81 *Picture is not a 1-D graph*

Error 81 means that you have used **xwires graph** on a display which is not a 1-D graph; probably you have simply mistyped the display number. (**xwires graph** only operates on an existing graph; if you have none initially, use something like **create 1 size 500,1 value 0; min=0 max=30; display preset** to create one).

82 *First point beyond right edge of graph*

Error 82 means that the first point you marked in a **xwires graph** command was unsatisfactory. Draw graphs from the left hand side to the right.

83 *Bad zoom factor: zoom factor*

Error 83 means that you have used an illegal zoom factor for display viewing – particularly a negative or zero value.

84 *Command with too many keys: command*

85 *Command with too many options: command*

86 *Command with too many open specifications: command*

Errors 84, 85 and 86 should not normally occur; they indicate a fault in the System Generation program *SEMGEN*, which should ensure that no verb descriptor can have more keys, options or open requests than can be accommodated by the interpreter's internal tables. Contact *Synoptics* for advice.

87 *Label for picture picture number not accessible*

Unless it is an indirect consequence of disc hardware failure, error 87 only occurs on tape devices in installations unable to backspace their tapes, and therefore unable to read a tape picture label more than once during a session. In such circumstances, even commands like **copy tape:n to 1** lead to the error when Semper attempts to transfer the title from source to output. (**show devices** reports a *nobackspace* flag if your installation cannot backspace tapes).

88 *Sub-region specified with more than one layer*

Error 88 means that you have attempted to use a rotated, resampled or skewed subregion with more than one layer. (Semper currently only supports 2-D operations on such regions).

89 *Display sub-region outside border limits*

Error 89 means that you have referred to a display (picture, partition or frame) subregion that is entirely outside the display. Likely causes include:

- mistyping the value of a **position** key
- confusing one partition with another (be explicit, e.g. **erase fs:4 ..** if in doubt)

show partition n reports the size and position of partition *n*.

Semper 6 Command Reference

90 *Device device number is not a picture storage device*

Error 90 means that you have tried to apply a picture processing or inspection operation, or simply **examine** etc., to a device that contains text (e.g. a *Help Library*) rather than pictures.

Likely causes of error 90 include:

- mistyping a device number
- using **examine** on a *Help Library* (use **help/topics** instead)

(**show devices** tells you about all the devices you have currently assigned.)

91 *Arithmetic underflow detected*

Error 91 means that an arithmetic underflow has been detected. This means that the result of a calculation is too small to be represented accurately. If you are using a PC, the value zero will be substituted as the result of the calculation. If you are using a VAX, this error is only produced if Semper has been compiled with Underflow checking switched on.

Likely causes include:

- multiplying together pictures with very small (but non-zero) values
- dividing a very small valued picture by very large valued one
- taking too large a negative power, for example, **a=-1e10 b=2^a**
- using a form *fp* picture you have **created** but not yet filled with any particular values
- using a half-processed output picture following an abandoned operation

92 *Bad floating-point value detected*

If you are using a PC, error 92 means that the Numeric Data Processor has found an unrepresentable value (infinity, 0/0 etc.), and has substituted the value zero instead. Likely causes include:

- using a form *fp* picture you have **created** but not yet filled with any particular values
- using a half-processed output picture following an abandoned operation
- taking too high a power, e.g. **a=1e10 b=2^a**

If you are using a VAX, error 92 means that Semper has met an illegal floating-point value when processing a picture. Likely causes include:

- using a form *fp* or *complex* picture you have created but not yet filled with any particular values
- using a half-processed output picture following an abandoned operation

Otherwise, error 92 means that a calculation has used or created an unrepresentable value (infinity, 0/0, etc.). Likely causes include:

- using a form *fp* picture you have **created** but not yet filled with any particular values
- using a half-processed output picture following an abandoned operation

Semper 6 Command Reference

93 *Arithmetic overflow detected*

Error 93 means that an arithmetic overflow has been detected. This means that the result of a calculation is too large to be represented accurately. If you are using a PC, the value zero will be substituted as the result of the calculation. If you are using a VAX, this error is only produced if Semper has been compiled with Overflow checking switched on. Likely causes include:

- multiplying together pictures with very large values
- dividing a very large valued picture by very small valued one
- using a form *fp* picture you have created but not yet filled with any particular values
- using a part-processed output picture following an abandoned operation
- if you are using a Sprynt system, storing to a smaller data class when the values do not fit (e.g. *fp* to *byte* when the range exceeds 0 to 255).

94 *Divide by zero detected*

Error 94 means that an attempt to divide by zero has been detected. This error is rare, as Semper spots the problem in advance in most cases, and reports error 95 instead. This error may occur in locally written code.

95 *Attempt to divide by zero*

Error 95 means that the divisor in an arithmetic operation was found to be zero. For example: $a=0$ $b=1/a$ or calculate $:2/:3$ where picture 3 contains some zero pixels.

96 *Attempt to take square root of negative value*

Error 96 means that the argument to *root* was found to be negative. For example: $a=-1$ $b=\text{root}(a)$ or calculate $\text{In}(:9)$ where picture 9 contains some negative pixels.

97 *Argument to LN negative or zero*

Error 97 means that the argument to *ln* was not positive. For example $a=-1$ $b=\text{ln}(a)$ or calculate $\text{root}(:9)$ where picture 9 contains some negative or zero pixels.

98 *Argument to ACOS or ASIN out of range -1 to 1*

Error 98 means that the argument to *ACOS* or *ASIN* did not lie in the range -1 to 1. For example: $a=-\pi$ $b=\text{acos}(a)$ or calculate $\text{asin}(:4)$ where picture 4 contains some pixels with a value greater than 1 or less than -1.

99 *Attempt to take fractional power of zero or negative value*

Error 99 means that an attempt to raise a negative value to a non-integer power was detected. For example: $a=-\pi$ $b=1.5$ $c=a^b$ or calculate $(:4^.5)$ where picture 4 contains some negative pixels

Semper 6 Command Reference

100 *No free directory slots on device device number*

Error 100 means that the directory on a program library has filled. Try **compressing** the device if **directory** indicates deleted slots. Otherwise create a new program library with more slots (use the **slots** keyword in **assign new**) and copy the programs across to the new library.

101 *Insufficient free space on device device number*

Error 101 means that the text area on a program library has filled. Try **compressing** the device if **directory** indicates deleted space. Otherwise create a new program library with more space (use the **size** keyword in **assign new**) and copy the programs across to the new library.

102 *Program Index overflow – reduce complexity*

Error 102 indicates that a program has more **for**, **loop** commands and labels than can be buffered internally. Either restructure the program to remove unnecessary labels, or split out part of it as a new program which can then be invoked from the remainder of the original.

103 *Local variables not allowed in FOR loop*

Error 103 means that you have a **local** statement directly inside a **for** loop. **local** statements are allowed inside programs called from within a **for** loop, but must be outside of any **for** loops within that program.

104 *Too many local variables, can't save variable*

Error 104 means that the local variable table has overflowed (see **show system** for the size of the table). Local variables are used when keys and options are quoted in a command, when variables are used as **for** loop variables, when explicit **local** variables are declared and also occasionally in code within some processing commands.

105 *LOOP doesn't match active FOR loop*

Error 105 means that the variable name specified in a **loop** command does not match the loop variable name of the inner-most active **for** loop. Likely causes for this error are:

- having a **for** loop in a macro accessed from a program
- having a **loop** command with the wrong loop variable name in a conditional expression.

(You should use **next** or **break** to conditionally affect **for** loops)

106 *FOR with no matching LOOP*

Error 106 means the interpreter cannot find a matching **loop** for a **for** command. This is more likely in a program, as you will be prompted for terminating **loops** interactively. Other possible causes are:

- using the wrong loop variable name, for example: **for i 1,10 ... loop j**
- having a **for** loop in a macro accessed from a program

Semper 6 Command Reference

107 *RUN not allowed from within a program or FOR loop*

Error 107 means you have a **run** command within a program or **for** loop. The **run** command is used to select input from a new source, as a side-effect the current command line is lost. Any **for** loop or active program would then be left in limbo. Include the contents of the run file as a program and use **library** instead.

108 *Program nest level exceeded*

Error 108 means you have exceeded the maximum level of nesting of **library** commands that are allowed. One possible cause of this is unterminated recursion, either directly (*lib xxx* within program *xxx*) or indirectly (*lib xxx* within program *yyy* and *lib yyy* within program *xxx*). Look at the error context messages to find out the likely cause. See **show system** for the local maximum level of nesting.

109 *FOR loop nest level exceeded*

Error 109 means you have exceeded the maximum level of nesting for a **for** loop that is allowed. See **show system** for the local maximum level of nesting.

110 *NEXT or BREAK doesn't match any active FOR loops*

Error 110 means that a **next** or **break** command has been encountered with an index variable that is not in use by any active **for** loop, or there are no active **for** loops at all.

111 *LOOP with no FOR*

Error 111 means a **loop** command has been encountered when there was no active **for** loops.

112 *Label label name is inside a FOR loop*

Error 112 means that you have tried to **jump** to a label that is within a **for** loop that is not currently active.

113 *Program device device number is in use*

Error 113 means that you have attempted to **deassign** or **reinitialise** a device that has an active program on it. This error will only occur within a program. Look at the error context to determine which program is active on the device.

114 *No help library assigned*

Error 114 means that no help library has been assigned, which means that no help information can be provided on specified topics. The command **help** by itself prints some text which explains how to use the command and describes other Semper commands that can provide information about your Semper session.

Semper 6 Command Reference

115 *No help for specified topics*

Error 115 means that no topics matching the keywords following the **help** command were found in any of the assigned help libraries. Type **help/topics** for a list of available topics.

116 *Device or picture number out-of-range*

Error 116 means that the value of a numerical expression either before or after the colon ':' operator is out of range. Device numbers must lie in the range 1 to the maximum allowed for the system (type **show system** to find out what is the maximum). Picture numbers must lie in the range 1 to 999.

117 *Help libraries can only be created by the Help Manager*

Error 117 means that you are trying to create a new help library within Semper. Help libraries can only be created outside Semper using the *Help Manager* utility (usually called by **helpman**).

118 *Display device, device number, is already assigned*

Error 118 occurs when you attempt to **reassign** a display device that is already assigned. If you want to change the characteristics of a display you must **deassign** it first. Currently the display device is forced to be device 1, but this will change in future versions.

119 *Invalid or unknown file type*

Error 119 usually occurs when an attempt is made to assign a file that is not a Semper disc file, for example, attempting to assign a text file.

120 *Reassignment of new disc files only allowed in full version*

This message only occurs in a demonstration version of Semper software. A more limited range of Semper operations are allowed in this context.

121 *Device device number is already assigned*

Error 121 occurs when you assign a device that is already in use. Type **show devices** to list the currently assigned devices.

122 *Value must be given for keyword/variable*

Error 122 means that some of the options or keywords given to a command require that another keyword or variable should also be given a value.

Semper 6 Command Reference

123 *Unable to create workspace device (retry count exceeded)*

Error 123 means that Semper has been unable to create a unique workspace area after 20 retries. Possible reasons include:

- system clock failure
- too many Semper users are trying to create workspace devices at the same time (networked/Multiuser systems only)
- the primitive routine *WAITS* is not performing correctly
- the primitive routine *MCTIME* is not performing correctly
- the primitive routine *MCDC61* is not performing correctly

124 *Device device number is not a text file*

This error means that a picture device, display, tape device or help library has been used in a context that requires a program library device.

125 *Program program name not found*

This error means that the specified program is not available in any of the currently assigned program library devices. Use **show programs** to produce a list of available programs.

126 *Program program name already exists*

This means that an attempt has been made to **copy** or **rename** a program onto an existing program. If this is what you really want to do, delete the existing program first.

127 *Attempt to delete active program program name*

This error means that you have tried to delete a program while the program is still in use. This can happen if one program invokes another program that tries to delete the first one.

128 *Program program name has a corrupted index*

Error 128 should not normally happen, as it means that the program library index is corrupt or has been written back incorrectly. Possible causes include system crashes when adding a new program or incompatible versions of Semper on the same host machine.

129 *Fortran i/o error number on unit number file filename*

Error 129 can be produced by any command that accesses the file input/output system. Look up the i/o error number in your local Fortran IOSTAT or run-time i/o error tables. On some systems Semper also produces a more meaningful interpretation of the error number as a supplementary message.

Semper 6 Command Reference

130 *File filename not found*

Error 130 means that the indicated file could not be found and/or opened for input. On some systems the local PATH environment string is used to locate input files, and this can result in unreadable versions of files being found before the intended file. In these conditions give the complete local pathname for the file.

131 *Unexpected end-of-file found in filename*

Error 131 occurs when a data file ends unexpectedly. This can happen when using the commands **read**, **input** etc if the original file was incorrectly written or in the wrong format.

132 *Picture number has no title*

Error 132 means that a command which requires a title for a picture was unable to locate one.

133 *Program name name is invalid*

Error 133 means that a program name contains some invalid characters. Valid characters are a to z, 0 to 9 and \$.

134 *Too many files open*

Error 134 means that the local Semper i/o system has run out of file handles. This is usually caused by opening an excessive number of files.

135 *File filename already exists*

Error 135 occurs when a command that outputs a file finds that the file already exists. In general these commands require the option **new** to overwrite a file.

136 *File name incorrectly specified*

Error 136 means that you have incorrectly specified a filename. Check for overlong names or invalid characters.

137 *Internal file name too long*

This error occurs when an internal buffer has overflowed when trying to construct a file name. Possible causes include:

- An error in the specification of the PATH in the local environment
- The current directory is at the end of a very long chain of directories
- The user specifying a very long file name component

138 *No match found*

This error means that the **examine** command cannot find a match in the directory for the range, text, class or form combination that you have specified.

Semper 6 Command Reference

139 *Command line buffer overflow*

This error message means that an 'immediate' command, such as **textfield execute**, has caused the delayed command line buffer to overflow. This error is usually caused by a user-written routine making excessive use of the Semper function OBEYCL without returning to the command interpreter.

140 *Security device missing*

This error message means that the required security device is missing or inoperable. Check that the device has not been accidentally removed.

141 *Wrong level security device*

This error message indicates that the security device does not match the software release version. Ensure that the correct device is connected.

142 , 143, 144 *Security failure*

This message means that a security problem has occurred (for example, a security device malfunction or violation). If you cannot locate the source of the problem please send a Customer Report form to *Synoptics* or contact your technical support service.

145 *LP lp number already opened*

This error indicates a syntax error in the command syntax file. There is more than one open statement referring to the specified LP number. Modify the syntax entry and rebuild Semper.

146 *Attempt to reference LP lp number before opening*

This error indicates a syntax error in the command syntax file. An open statement for a new image references an unopened LP in the comparison field (e.g. open (lp2,new,lp1) where there is no open for LP1). Modify the syntax entry and rebuild Semper.

147 *Picture picture number has bad value value in label field field name*

This error should not normally happen in Semper. It is a less severe form of error 52, and indicates that one or more crucial fields in an existing picture label have corrupt values. The first corrupt field found and the erroneous value are reported in the error message. As with error 52 the only really useful response is to delete the picture.

Semper 6 Command Reference

148 **Picture picture number new value value In label field field name Is invalid**

This error indicates an attempt to write a picture label that has one or more crucial fields with values that are out of range (e.g. dimensions with zero size, invalid picture form). This can happen when attempting to import external data files that are in the wrong format or which have an incorrectly produced Semper label in them. If this is the case the image may still be recoverable by indentifying the field Semper is complaining about (given in the error message) and correcting the value – either by correcting the program sourcing the file or by patching the file with a suitable text or disc editing utility.

149 **Attempt to write a malformed label for picture picture number**

Error 149 indicates an attempt to write a totally corrupt picture label to disc. This can happen when attempting to import external data files that are in the wrong format. It can also happen when user written code overwrites the label data before writing it out to disc.

150 ***Chord buffer overflow***

Error 150 means that there are too many particles intersected by a picture row. This could be caused by too much noise in the source image. Some additional processing to clean up the source image or adjustment of the threshold values given to **analyse** might help to reduce the number of spurious particles. If, however, all the particles are genuine, the last resort is to break down the source image into several smaller pictures.

151 ***Too many particles***

Error 151 means that the particle parameter list is not big enough to hold the results for all the particles found by **analyse**. The size of the particle parameter list is limited by the row buffer size. The **area** key can be used to screen out particles less than a certain size. **analyse** can also be made to output the results for only those particles which fall within a specified subregion of the source picture (using the 2-D subregion keys when invoking **analyse**).

Error number 152 is currently not in use

153 ***Too many items to label in segmented picture***

Error 153 will occur if a byte-formatted segmented picture is requested and there is too much detail in the source image. No more than 255 particles can be recorded in a byte-formatted segmented picture. Depending on the complexity of the particles, this limit may be reduced during the scan through the source image.

154 ***Bad pixel value in segmented picture***

Error.154 means that a negative pixel value has been encountered in a segmented picture. A segmented picture should contain zero (background) and positive (particle id) pixel values.

155 ***Particle not found***

Error 155 means that the specified particle id was not found in the particle parameter list.

Semper 6 Command Reference

156 *Excessively large curve*

Error 156 means that the coordinate limits of the curve supplied to **pcurve** exceed the range -32000 to 32000.

157 *Picture is not a 2-D image*

Error 157 means that the operation to be carried out with the specified display picture requires a 2-D image.

158 *No particles selected*

Error 158 means that none of the particles described in the particle parameter list satisfy the constraints given by the **if** and **unless** keys.

159 *Zero pixel area*

Error 159 means that there were no pixel positions generated inside the specified closed curve. This means that results that depend on summing values at all the pixel positions, for example, total intensity, are not defined.

160 *Particle not present in segmented picture*

Error 160 means that there are no pixels in the segmented picture set to the specified particle id.

161 *Event queue routine error*

Error 161 is an internal error that indicates problems in the low level event interface routines. If the error persists please submit a Report Form to *Synoptics* (see *Appendix I, Customer Report Form*).

162 *Device device number already quoted*

Error 162 means that a command that expects a list of devices (for example, **order**) has found a particular device used more than once in the list.

163 *Look-up table number not currently active*

This message means that a command requires an active lookup table (for example, **lut keys**) and did not find one.

164 *CTF wave requires full-plane Fourier pictures*

This message means that the command **ctf** with the option **wave** was given a half-plane Fourier picture as a source.

Semper 6 Command Reference

165 *Attempt to use invalid LP number lp number*

Error 165 means that a call to a kernel function (e.g. SEMROW) has been made with an LP number that is out of the valid range 1 to NLPS.

This usually indicates an error in locally written code, for instance passing the wrong variable as the LP number.

166 *Attempt to use unopened LP number lp number*

Error 166 means that a call to a kernel function (e.g. SEMROW) has been made with an LP number that does not refer to an open picture.

This usually indicates an error in locally written code, for instance passing the wrong variable as the LP number or failing to open a picture before accessing it. Check that any implicit OPENS that the local code expects are present in the command descriptor.

Error numbers 167 to 191 are currently not in use

192 *FORTTRAN subscript out of range (program fault)*

This error indicates that a *subscript out of range* error occurred in a FORTRAN routine. If the problem is not caused by a routine of your own writing, please send a Customer Report form to *Synoptics*.

193 *FORTTRAN adjustable array dimension error (program fault)*

This error message occurs when an adjustable array dimension, passed to a FORTRAN routine, is zero or negative. For example:

```
CALL SUB (A, 0)
.
.
.
SUBROUTINE SUB (A, N)
INTEGER N
REAL A (N)
```

If this error has not been caused by a user-written routine, please send a Customer Report Form to *Synoptics*.

Error numbers 194 to 249 are currently not in use

250 *No UIF error – status set TRUE with no error code*

You should not normally get this error but you might get it when performing operations on elements with the variable *eno* set incorrectly. Has the **create** option been omitted?

Semper 6 Command Reference

251 *UIF is not initialised*

Error 251 means that you have issued a user interface command before giving the **uif enable** command. This command initialises internal variables.

252 *Maximum number of windows exceeded*

Error 252 means that that there are too many windows. A window is a *panel*—so you are using too many panels. Use the **uif status** command for a list of the number of allowed panels.

253 *Maximum number of panels exceeded*

Error 253 means that that there are too many panels. Use the **uif status** command to see a list of the available number of panels. Note that pull-down and pop-up menus use up (hidden) panels.

254 *Maximum number of scrolling areas exceeded*

Reserved for future use.

255 *Maximum number of cells exceeded*

Error 255 means that that there are too many cells. Use the **uif status** command to list the available number of cells.

256 *Maximum number of menus exceeded*

Error 256 means that that there are too many menus. Use the **uif status** command to list the available number of menus.

257 *Maximum number of textfields exceeded*

Error 257 means that that there are too many textfields. Use the **uif status** command to list the available number of textfields.

258 *Operation cannot be performed while panel is showing*

Error 258 means that you are trying to perform an operation when a panel is visible. This may be caused by:

- an attempt to add a new element to the panel
- an attempt to alter any of the characteristics of the panel

259 *Operation cannot be performed while scrolling area is showing*

Reserved for future use.

260 *Operation cannot be performed while panel is not showing*

Error 260 means that the operation cannot be performed while the panel is showing, because the operation would cause the size of the panel to change.

Semper 6 Command Reference

261 *Error setting variable value*

Error 261 occurs when the user interface system is setting a value to a variable. This is most likely if you have used too many variables (try **unset** some of them). If not, this indicates an internal error. Contact *Synoptics* for advice.

262 *Error getting variable value*

An internal error, please report this to *Synoptics*.

263 *Error creating window*

An internal error, please report this to *Synoptics*.

264 *Error clearing window*

An internal error, please report this to *Synoptics*.

265 *Error obtaining size of window device*

An internal error, please report this to *Synoptics*.

266 *Error showing window*

An internal error, please report this to *Synoptics*.

267 *Error destroying window*

An internal error, please report this to *Synoptics*.

268 *Error hiding window*

An internal error, please report this to *Synoptics*.

269 *Error moving window*

An internal error, please report this to *Synoptics*.

270 *Error creating panel*

An internal error, please report this to *Synoptics*.

271 *Error destroying panel*

An internal error, please report this to *Synoptics*.

272 *Error hiding panel*

An internal error, please report this to *Synoptics*.

Semper 6 Command Reference

273 *Error showing panel*

Error 273 means that the panel is too big to display on the device (framestore/host screen) or the panel is located outside the display limits. Very often, this is because **justification** is incorrectly set.

274 *Error moving panel*

Error 274 means that an attempt was made to move the panel outside the device limits.

275 *Error naming panel*

An internal error, please report this to *Synoptics*.

276 *Error setting panel/element colours*

You are trying to set a colour that your hardware cannot support.

277 *Error clearing device*

An internal error, please report this to *Synoptics*.

278 *Attempt to perform operation when mandatory panel showing*

Error 278 means that a *mandatory* panel is showing but you tried to perform some other user interface related action, for example showing another panel or popping up a menu. Remember that when a mandatory panel is showing interactions are only allowed with it or elements on it.

279 *Invalid panel identifier*

Error 279 means that you have used an identifier for a panel which the user interface system knows nothing about. Are you specifying the Id key?

280 *Invalid scrolling area identifier*

Reserved for future use.

281 *Error showing scrolling area*

Reserved for future use.

282 *Invalid position*

Error 282 means that an attempt has been made to position the panel outside the limits of the device (framestore/host display).

Semper 6 Command Reference

283 *String too short for copy/concatenate operation*

An internal error, please report this to *Synoptics*.

284 *UIF Initialisation failed*

Error 284 means that the user interface system was unable to initialise. Have you already initialised the system? It is advisable to exit Semper and then re-start it.

285 *Invalid device requested*

Error 285 means that an attempt was made to use an unknown device for a panel. Is the variable *cdi* set correctly? Permitted values for *cdi* are 0, 1 and 2.

286 *Invalid element type*

An internal error, please report this to *Synoptics*.

287 *Invalid action*

An internal error, please report this to *Synoptics*.

288 *Invalid element position/size*

Error 288 means that the specified element (menu, cell, textfield) was too big or was badly positioned on its panel:

289 *Invalid cell identifier*

Error 289 means that you have used an identifier for a cell which the user interface system knows nothing about. Have you passed the right value with the *id* key? Is the variable *eno* set correctly? Have you omitted the **create** option?

290 *Invalid cell highlighting type*

An internal error, please report this to *Synoptics*.

291 *Invalid textfield identifier*

Error 291 means that you have used an identifier for a textfield which the user interface system knows nothing about. Have you passed the right value with the *id* key? Is the variable *eno* set correctly? Have you omitted the **create** option?

292 *Invalid textfield length*

Error 292 means that you have given an incorrect length for a textfield. Textfield lengths should be greater than zero and less than the amount of (dynamic) memory available. Type **uif status** for a list of the available memory.

Semper 6 Command Reference

293 *Invalid numeric textfield range specifier*

Reserved for future use.

294 *Numeric textfield contents out of range*

Reserved for future use.

295 *Unable to convert numeric textfield contents to a number*

Reserved for future use.

296 *Unable to convert number to a string*

Error 296 means that Semper was unable to convert the string given into a number.

297 *Invalid menu identifier*

Error 297 means that you have used an identifier for a menu which the user interface system knows nothing about. Have you passed the right value with the **id** key? Is the variable *eno* set correctly? Have you omitted the **create** option?

298 *Invalid menu type*

An internal error, please report this to *Synoptics*.

299 *Invalid menu style*

An internal error, please report this to *Synoptics*.

300 *Menu panel has not been created*

An internal error, please report this to *Synoptics*.

301 *Action string is too long to fit application input buffer*

Error 301 means that the string that you specified for an action is too long to fit into Semper's command line buffer. An action might be using the **begins** key action on a cell, or the click of a left mouse button, or the execution of a textfield. Try breaking the command down into smaller parts.

302 *Framestore access error*

Error 302 means that while the user interface system was accessing the framestore, the framestore reported an error. Check for framestore hardware failure.

303 *UIF is not running*

An internal error, please report this to *Synoptics*.

Semper 6 Command Reference

304 *Window of requested size/position will not fit device*

Error 273 means that the panel is too big to display on the device (framestore/host screen) or the panel is located outside the display limits. Very often, this is because **justification** is incorrectly set.

305 *UIF termination failed*

An internal error, please report this to *Synoptics*.

306 *Cursor position stack is empty*

Error 306 means that **device restore** has been used before **device save** has been used (or **restore** has been used more times than **saved** is used).

307 *Cursor position stack is full*

Error 307 means that you have given too many **device save** commands. The cursor position stack is of a fixed size, typically about four positions. Have you omitted to restore a position somewhere?

308 *Dynamic memory system not initialised*

An internal error, please report this to *Synoptics*.

309 *Dynamic memory system already initialised*

An internal error, please report this to *Synoptics*.

310 *Invalid block size of dynamic memory requested*

An internal error, please report this to *Synoptics*.

311 *Invalid logical index to dynamic memory used*

An internal error, please report this to *Synoptics*.

312 *Dynamic memory logical index table full*

An internal error, please report this to *Synoptics*.

313 *Dynamic memory exhausted*

Error 313 means that all the memory reserved for storing names of objects (panels, elements), actions, etc. has been used. **ulf status** will show how much you have left. The amount of memory available is fixed and installation dependent.

Semper 6 Command Reference

314 *Invalid positioning point*

An internal error, please report this to *Synoptics*.

315 *Invalid mouse button number*

An internal error, please report this to *Synoptics*.

316 *Display is not assigned for windows on framestore*

Error 316 means that you have tried to show a panel on a display that has not yet been assigned.

317 *Invalid element identifier*

An internal error, please report this to *Synoptics*.

318 *Invalid panel/scrolling area/element identifier*

An internal error, please report this to *Synoptics*.

Error numbers 319 to 332 are currently not in use

333 *The Semper UIF is not installed in this system*

UIF is not installed in this system and you have tried to use a UIF facility or command. The Semper system is either a BlackBox version of Semper, or has had the User InterFace code explicitly removed to reduce code size and memory requirements. Note that it is not possible to run the TUTOR interface without having UIF in Semper.

Error numbers 334 to 499 are currently not in use

500 *Source and covariance sizes differ*

The number of layers of source picture must correspond with the number of columns of the covariance matrix.

501 *Too many classification regions*

The number of classification regions that can be processed by the classification commands (**box**, **mindistance** and **likelihood**) is dependent on the length of a Semper row buffer. The more layers a picture has, the fewer classification regions it may have.

502 *Too many layers*

The number of layers that can be processed by the classification commands depends on both the length of a Semper row buffer and on the width of the picture. The wider a picture is, the fewer layers it may have.

Semper 6 Command Reference

503 *Threshold must be in the range 0.0 .. 100.0*

Thresholds to the **likelihood** command, maximum likelihood, are given in terms of percentages. The percentage gives the number of pixels in the class which will be classified. For this reason the percentage must be less than 100.0.

504 *Not enough thresholds*

When more than one threshold is given there must be as many thresholds as classes.

505 *Probabilities must be positive, less than 1.0*

Probabilities must be in the range 0.0 (never) to 1.0 (always).

506 *Sum of probabilities greater than 1.0*

The sum of probabilities for all classes (in the **likelihood** command) must sum to less than 1.0.

507 *Too many channels, maximum (square) is: number*

In order to achieve a respectable performance the 2-D histogram is built up in memory. As with other Semper commands the maximum number of channels is proportional to the length of a Semper row buffer.

508 *Thresholds must be positive*

Since thresholds are in terms of standard deviations it is not sensible to give negative thresholds.

509 *Unable to calculate eigenmatrix*

The maximum iteration count was reached in calculating the eigenvectors. This should not arise with covariance matrices calculated with the **covar** command. If the covariance matrix has been determined by some other method ensure it is symmetric, as the solution (*Jacobi's* method) relies upon this.

510 *Invalid polynomial order, largest is: number*

The maximum order polynomial is restricted by the (implementation dependent) size of a Semper row buffer. The polynomial order must be greater than zero.

511 *Control points must be plists*

The control points must be supplied in *Plist* class pictures (perhaps use the **reclass** command to do this).

Semper 6 Command Reference

512 *Image and map plists must be the same length*

The map and image plists (control points) must be of the same length. Otherwise the **warp** command cannot perform point matching.

513 *Too many control points, maximum is: number*

The **warp** command can only handle a certain number of control points. This should, in normal circumstances, be sufficient to solve the maximum order of polynomial allowed. Use the **cut** command to reduce the size of the list.

514 *Not enough control points (or singular set)*

When solving the equations for the coefficients not enough points are in the control data set to solve the equations. Alternatively, the set of equations is ill-conditioned, so it is not possible to solve them. Try adding more control points.

515 *Number of classes and probabilities disagree*

The number of probabilities given must be equal to the number of classes.

516 *Class has zero determinant – no inverse possible*

The covariance matrix is singular and it is not possible to evaluate its inverse. This, in turn, means that the maximum likelihood method cannot be used.

Error numbers 517 to 899 are currently not in use

900 to 949 Message depends on error condition

This is a non-specific error generated by a user-supplied command. In general, any non-standard errors generated by routines not supplied by *Synoptics* should be in the range 900 to 949.

Error numbers 950 to 996 are currently not in use

997 QUIT in response to page prompt

This error means that a *QUIT* option was selected in response to the page prompt. Strictly speaking, this is not an error in that it generates no error message or traceback and diagnostic information, but it is a condition which will cause commands and programs to abort. For this reason, it can be trapped like a normal error. Note that you can disable this condition using the **page...noquit** command to disable the *QUIT* facility.

998 Internal error number status number routine name

This is a non-specific internal Semper error. It is used to flag problems that may occur at a low level within Semper. If you encounter this error, please make a note of the information that it reports and contact *Synoptics*.

Semper 6 Command Reference

999 *FATAL error number status number routine name*

This error is a "fatal" error and means that Semper cannot continue and will shut itself down. This occurs if Semper encounters a condition which means that the session can no longer continue, for example, if commands can no longer be read from the terminal. Semper attempts to close the session in an orderly fashion, as if you had typed **stop**. If you see this error, please make a note of the information that it reports and contact *Synoptics*.

Please note that error numbers 900 to 949 are reserved for Fortran errors from user-written routines. Refer to the manual:

Fortran Programmers' Guide

for details of how to implement these errors.

Appendix F

PROTECTED & FIXED

VARIABLES

Overview

This appendix describes the small number of Semper variables that have special meanings and properties. Variables have a name and an associated numerical value. They are used to hold and manipulate values for various purposes. Semper maintains a list of the variables that are set at any given time. A variable that is not on this list has no value at all and is called *unset*. Some special Semper variables on this list are called *protected* or *fixed* variables and are described below.

Protected variables

The following variables are *protected* variables, which means that you cannot change their values by assignment:

<i>ric</i>	release identification code (for example, 6.2)
<i>yes</i>	=1, used as logical value <i>true</i> and for answering questions
<i>no</i>	=0, used as logical value <i>false</i> and for answering questions
<i>pi</i>	= π = 3.1415927
<i>select</i>	the current picture number
<i>display</i>	the current display number
<i>rc</i>	error number after a command is executed
<i>cframe</i>	current display frame number
<i>clut</i>	current look-up-table number

Fixed variables

The following variables are *fixed* variables, which means that you can change their values by assignment but cannot unset them:

<i>min, max, mean, me2, sd</i>	last determined pixel data range, mean, standard deviation
<i>trap</i>	used in user controlled error handling
<i>cd</i>	current picture device number
<i>fs</i>	current display device number

Appendix G

PIXEL

CONNECTIVITY

Overview

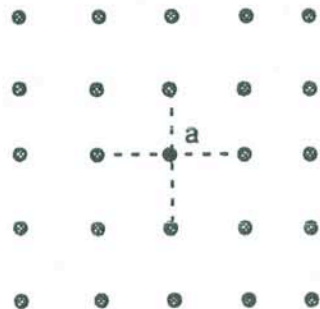
This appendix explains the concept of *pixel connectivity* that is used by the morphology commands **analyse**, **erode**, **dilate** and **median**. Pixel connectivity determines how pixels are grouped together to form objects.

4 and 8 connectivity

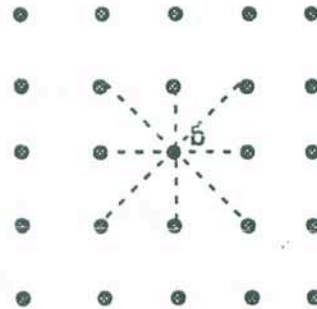
There are two kinds of pixel connectivity used:

- 4-connectivity
- 8-connectivity

In 4-connectivity, pixels are treated as if they are connected to their neighbours horizontally and vertically, but not diagonally. In 8-connectivity, diagonal neighbours are also treated as connected as shown in the diagram below. Pixel *a* is linked by 4-connectivity to four of its neighbours. Pixel *b* is linked by 8-connectivity to all eight of its neighbours.



4-connectivity



8-connectivity

Semper 6 Command Reference

For example, the following group of pixels forms a single region using 8-connectivity but forms four separate regions using 4-connectivity.



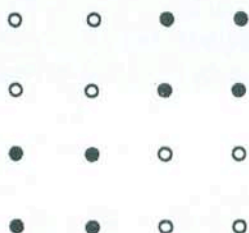
*pixel links using
8- connectivity*



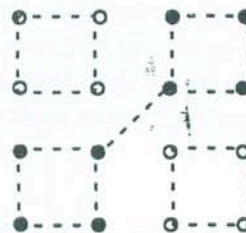
*pixel links using
4- connectivity*

Object and background connectivity

8-connectivity gives smoother curves and better rendering of edges, however Semper does not use it for both shapes and background as this is topologically inconsistent. For example, both object and background are 8-connected across the centre of the configuration given below. Semper therefore uses 8-connectivity to link objects and 4-connectivity for the background to separate the shape from the background. Object pixels are shown in black, background pixels in white.



original configuration



*8-connectivity used to link object,
4-connectivity for background*

This type of object and background separation is used, for example:

- by the **analyse** command to distinguish a particle from its surroundings
- by the **dilate separately** command to shrink the background to a 4-connected skeleton to maintain separation of objects
- by **erode skeleton** to produce an 8-connected skeleton

Appendix H

ILLUMINATION

Overview

The **sheet** and **solid** commands make use of an illumination or lighting model to determine the light intensity at every point of the shaded surface representation. There are four components in the lighting model:

- ambient lighting
 - depth contrast
 - forward lighting
 - directed lighting
- } diffuse and specular reflections

This appendix describes each of these components and presents the lighting equation which specifies exactly how these components are combined.

Ambient lighting

Ambient light is non-directional and is applied uniformly at all points of a surface that it illuminates. The amount of ambient light is specified by the **ambient** key (default=40). It is used to light all parts of an object so that the object stands out from the background.

Depth contrast

Depth contrast (or depth cueing) makes parts of an object that are further away from you less bright. This conveys some sense of depth for an object. The value that you specify with the **dcontrast** key is the difference in brightness between the front and rear of the object (default=30).

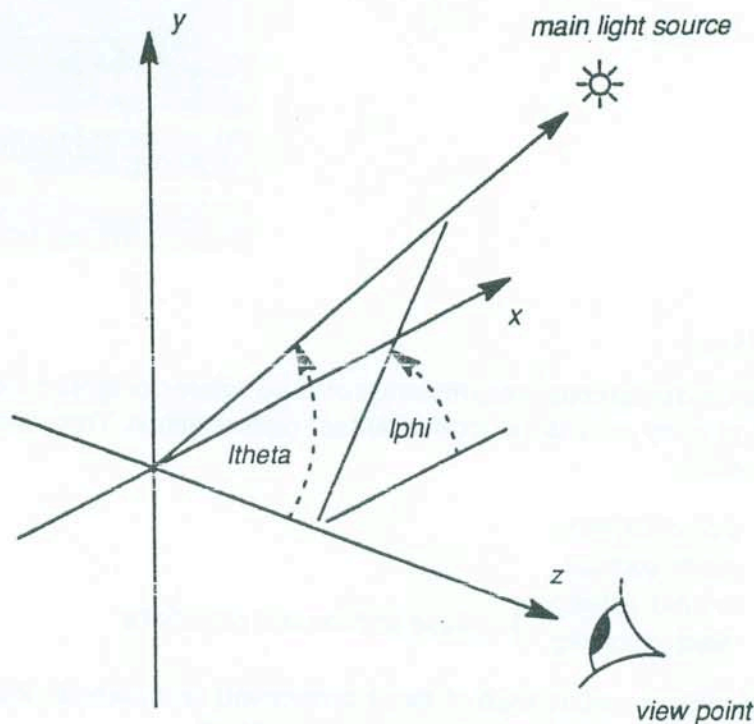
Forward lighting

This is a source of light that shines along the viewing direction. It helps to provide lighting on regions of a surface that are not illuminated from the main light source. If a surface is turned away from the eye, there can be no contribution from this light source, and in any case, the surface is not visible, so all lighting is suppressed. The intensity of the forward light source is specified with the **forward** key (default=42).

Directed lighting

The main light source is the only light source whose direction can be controlled. The keys **ltheta** and **lphi** specify the direction with respect to the viewing direction. The light source is inclined at an angle **ltheta** from the positive z axis, with an x-y azimuth of **lphi** anti-clockwise from the positive x axis. The axes mentioned here lie in the projected coordinate space, with the axes centred on the source origin and with the eye looking along the negative z-axis.

Semper 6 Command Reference



By default the light shines from your right ($ltheta=pi/4$, $lphi=0$). The intensity of the main light source is specified with the **main** key (default=154).

Diffuse and specular reflection

The light from the two light sources (forward and main) is reflected from a surface as two components. The **sdr** key specifies how the incident light intensity is to be split between diffuse and specular reflections (default=0.4). *Diffuse reflection* models the scattering of light at a surface in all directions and depends only on the incident angle of the light on the surface. *Specular reflection* models the reflection of light from a polished or mirror-like surface, where the reflected intensity is greatest when the surface normal is midway between the lighting and eye directions, and it falls off rapidly at other angles. It is specular reflection that generates well defined highlights on an object. The value of **sdr** is a measure of surface reflectivity – zero models a perfectly rough surface and large values (3 or more) model a highly polished surface.

Lighting equation

$$\begin{aligned} \text{surface intensity} = & \text{ambient} \\ & + \text{dcontrast} * z / (z_{\text{max}} - z_{\text{min}}) \\ & + \text{forward} * (d * n \cdot e + s * sr(n \cdot e)) \\ & + \text{main} * (d * n \cdot l + s * sr(n \cdot m)) \end{aligned}$$

where: z = surface z value
 z_{max} = maximum projected z value
 z_{min} = minimum projected z value

Semper 6 Command Reference

n = surface normal

e = eye direction

l = main light source direction

m = 'mirror' normal = $\frac{e+l}{|e+l|}$

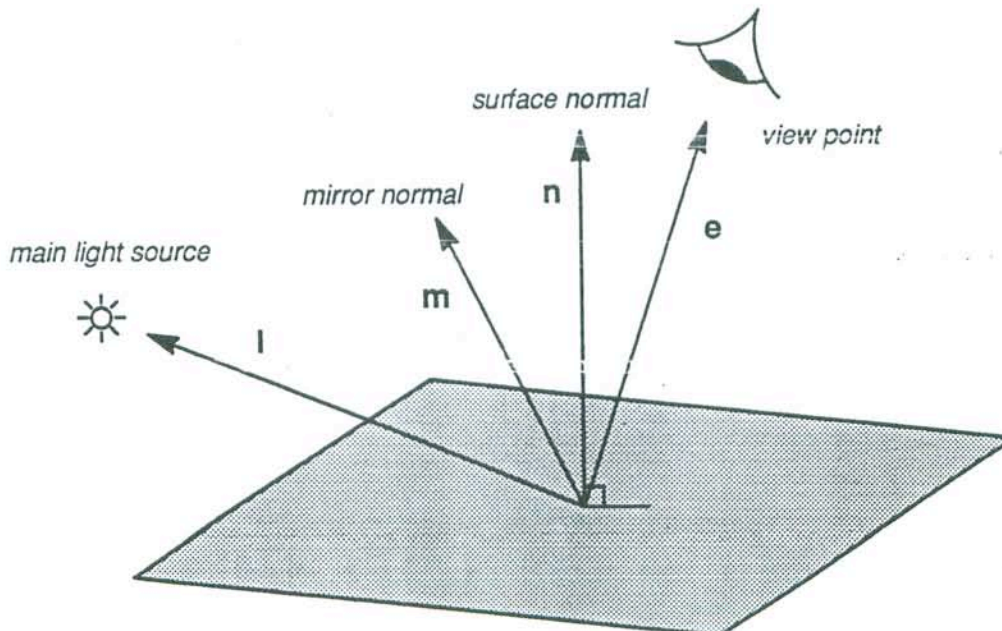
d = diffuse reflection factor = $\frac{1}{1+sdr}$

s = specular reflection factor = $\frac{sdr}{1+sdr}$

$sr(x)$ = specular reflection function = $\frac{0.03x}{1.03-x}$

Notes:

1. n , e , l , m are all unit vectors.
2. If $n \cdot e < 0$ the surface points away from the eye, so the surface intensity is zero.
3. If $n \cdot l < 0$ the main light source does not illuminate the surface, so its contribution is zero.



Appendix I

CUSTOMER REPORT

FORM

Using the report form

Please fill in and detach the form given overleaf if you find a fault with the following:

- software
- hardware
- documentation

Also use this form to record technical enquiries, enhancement requests, suggestions or comments.
Send the form to the appropriate address given below:

Synoptics Ltd
271 Cambridge Science Park
Milton Road
Cambridge
CB4 4WE
UK
Tel: (0223) 423223
Fax: (0223) 420020

Synoptics Ltd
400 Amherst Street
Nashua
New Hampshire 03063
USA
Tel: 603 881 7035
Fax: 603 881 7078

Appendix J

ASCII KEY CODES

Overview

This appendix describes the ASCII key codes that are used by Semper. Each keystroke from the keyboard is converted to an internal character code (see the commands **event** and **Inkey**). Semper uses the standard ASCII character sequence for the keycodes, with some additions for special keystrokes such as function keys and cursor keys. The ASCII codes and special keycodes are shown in the tables given below.

ASCII keycodes

space	32	0	48	@	64	P	80	'	96	p	112
!	33	1	49	A	65	Q	81	a	97	q	113
"	34	2	50	B	66	R	82	b	98	r	114
#	35	3	51	C	67	S	83	c	99	s	115
\$	36	4	52	D	68	T	84	d	100	t	116
%	37	5	53	E	69	U	85	e	101	u	117
&	38	6	54	F	70	V	86	f	102	v	118
'	39	7	55	G	71	W	87	g	103	w	119
(40	8	56	H	72	X	88	h	104	x	120
)	41	9	57	I	73	Y	89	i	105	y	121
*	42	:	58	J	74	Z	90	j	106	z	122
+	43	;	59	K	75	[91	k	107	{	123
,	44	<	60	L	76	\	92	l	108		124
-	45	=	61	M	77]	93	m	109	}	125
.	46	>	62	N	78	^	94	n	110	~	126
/	47	?	63	O	79	_	95	o	111		

Special keycodes

9	Tab	257	Backspace	513	Cursor up
13	Return	258	Delete line	514	Cursor down
27	Escape	259	Insert/replace mode	515	Cursor left
		260	Start of line (home)	516	Cursor down
		261	End of line		
		262	Refresh line		

768 - 1023 Function key= 768 + key number