Chapter 5.
Options

Topic: Ignore

Options may be set to modify the behavior of ACIS. An option's value may be a flag (indicating an on/off state), a number (integer or real number), or a string. Options may be set in a Scheme application (such as Scheme AIDE) using the Scheme extension option:set; in the ACIS Test Harness using the command option; or in a C++ application using one of several API functions. Refer to the *3D ACIS Online Help User's Guide* for a description of the fields in the reference template.

check_abort

Option: Modeler Control, Error Handling, Debugging

Action: Determines whether to abort entity checking.

Name String: **check_abort**

Scheme:	boolean	#f, #t	#f
Test Harness:	integer	0, 1	0
C++:	logical	FALSE, TRUE	FALSE

Description: If this option is on, entity checking aborts after the first check problem is found.

Example:
```
; check_abort
; Turn on check aborting
(option:set "check_abort" #t)
;; #f
```

check_level

Option: Modeler Control, Debugging, Error Handling

Action: Controls the amount of entity checking performed.

Name String: **check_level**

Scheme:	integer	10, 20, 30, 40, 50, 60, 70	20
Test Harness:	integer	10, 20, 30, 40, 50, 60, 70	20
C++:	int	10, 20, 30, 40, 50, 60, 70	20

Description: This option controls the amount of entity checking performed in the "sanity check" code.

The valid levels are defined with the enumerated type c_level. The different check levels are:

10 = Fast error checks
20 = Level 10 checks plus slower error checks
30 = Level 20 checks plus D-Cubed curve and surface checks
40 = Level 30 checks plus fast warning checks
50 = Level 40 checks plus slower warning checks
60 = Level 50 checks plus slow edge convexity change point checks
70 = Level 60 checks plus face/face intersection checks

Example:

```
; check_level
; Set for slower warning checks
(option:set "check_level" 50)
;; 20
```

cone_param_lines

Option: Modeler Control

Action: Enables or disables the display of parameter lines on cones in the Test Harness.

Name String: **cone_param_lines**

Scheme:	Not applicable		
Test Harness:	integer	0, 1	1
C++:	Not applicable		

Description: Turning on this option enables the display of cone parameter lines. The number of parameter lines is set using the options u_param_lines and v_param_lines. This option only applies to the Test Harness.

Example: Not applicable

cone_silhouettes

Option: Modeler Control

Action: Enables or disables display of a cone silhouette in the Test Harness.

Name String: **cone_silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: If on, enables display of a cone silhouette. Option silhouettes must also be on. This option only applies to the Test Harness.

Example: Not applicable

d3sil_approx_sf

Option: Modeler Control, Silhouette and Isoparametric Curves

Action: Determines whether to calculate the surface silhouette on the approximating surface.

Name String: **d3sil_approx_sf**

Scheme: boolean #f, #t #f

Test Harness: integer 0, 1 0

C++: logical FALSE, TRUE FALSE

Description: For spline surfaces, if this option is on, the surface silhouette is calculated on the approximating surface, in order to speed up the calculation. If this option is off, the surface silhouette is calculated on the true surface.

Example:

```
; d3sil_approx_sf
; Use the approximating surface
(option:set ”d3sil_approx_sf” #t)
;; #f
```

d3_checks

Option: Modeler Control, Debugging, Error Handling

Action: Evaluates D-Cubed geometric checks for legality of curves and surfaces.

Name String:	**d3_checks**		
Scheme:	boolean	#f, #t	#f
Test Harness:	integer	0, 1	0
C++:	logical	FALSE, TRUE	FALSE

Description: Enables the “D3” checks when turned on. This means that additional geometric checks will be performed on INTCURVEs and SPLINEs when api_check_entity is called. These checks are the same ones invoked by api_check_edge and api_check_face.

Example:
```
; d3_checks
; Turn on additional checks
(option:set ”d3_checks” #t)
;; #f
```

fuzzy_angle

Option: Modeler Control, Intersectors

Action: Sets the angle criterion for deciding if a nontangent intersection should be treated as fuzzy.

Name String:	**fuzzy_angle**	
Scheme:	real	0.5
Test Harness:	double	0.5
C++:	double	0.5

Description: For nontangent intersections, an angle criterion is used to decide whether to treat the intersection as “fuzzy.” If the dot product of the curve tangent and surface normal is less than this angle criterion, the intersection is called “fuzzy”. The value is somewhat arbitrary, but too high imposes too much overhead on simple cases, and too small causes too many near-coincidences to fail.

Example:
```
; fuzzy_angle
; Set the fuzzy angle a little higher
(option:set ”fuzzy_angle” .6)
;; 0.5
```

mesh_param_lines

Option: Modeler Control

Action: Enables or disables display of parameter lines on a mesh in the Test Harness.

Name String: **mesh_param_lines**

Scheme: Not applicable

Test Harness: integer 0, 1 0

C++: Not applicable

Description: This option only applies to the Test Harness.

Example: Not applicable

mesh_silhouettes

Option: Modeler Control

Action: Enables or disables display of a mesh silhouette in the Test Harness.

Name String: **mesh_silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: This option only applies to the Test Harness.

Example: Not applicable

plane_param_lines

Option: Modeler Control

Action: Enables or disables the display of parameter lines on planes in the Test Harness.

Name String: **plane_param_lines**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: When on, enables the display of parameter lines on planes. The number of parameter lines is set using the options u_param_lines and v_param_lines. This option only applies to the Test Harness.

Example: Not applicable

plane_silhouettes

Option: Modeler Control

Action: Enables or disables the display of plane silhouettes in the Test Harness.

Name String: **plane_silhouettes**

Scheme:	Not applicable		
Test Harness:	integer	0, 1	1
C++:	Not applicable		

Description: If on, enables display of a plane silhouette. Option silhouettes must also be on. This option only applies to the Test Harness.

Example: Not applicable

show_warning_msg

Option: Modeler Control, Silhouette and Isoparametric Curves

Action: Determines whether all the warning messages that the body checker generates should be added to the insanity list and displayed.

Name String: **show_warning_msg**

Scheme:	boolean	#f, #t	#f
Test Harness:	integer	0, 1	0
C++:	logical	FALSE, TRUE	FALSE

Description: In an optimized build this option is set to FALSE (its setting is TRUE in a debug build). Turning it on means that all warning messages that the body checker generates are added to the insanity list and are displayed. Some of the warnings indicate problems in the geometry that might cause difficulties during different modeling operations.

Example:

```
; show_warning_msg
; Set the warning messages display off
(option:set ”show_warning_msg” #f)
;; #t
```

sil_edges

Option: Modeler Control, Silhouette and Isoparametric Curves

Action: Controls whether silhouettes are calculated in the Test Harness using an API or low-level code.

Name String: **sil_edges**

Scheme: Not applicable

Test Harness:	integer	0, 1	0

C++: Not applicable

Description: This option only applies to the Test Harness. If this option is on (true), API api_silhouette_edges is used to calculate silhouettes. This has some overhead (the API makes edges), but it tests the API. If this option is off (false), silhouettes are calculated using lower level code (i.e., no API is called and edges are not made).

Example: Not applicable

silhouette_curves

Option: Modeler Control, Silhouette and Isoparametric Curves

Action: Sets the type of silhouette curves used.

Name String: **silhouette_curves**

Scheme:	boolean	#f, #t	#t
Test Harness:	integer	0, 1	1
C++:	logical	FALSE, TRUE	TRUE

Description: If on, instructs the D–Cubed silhouette algorithm to make para_sil_int_curs or persp_sil_int_curs as appropriate, instead of surf_int_curs.

Example:

```
; silhouette_curves
; Turn option off
(option:set "silhouette_curves" #f)
;; #t
```

silhouettes

Option: Modeler Control

Action: Enables or disables the display of silhouette lines on any surface type display of plane silhouettes in the Test Harness.

Name String: **silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: When on, enables the display of silhouette lines. Silhouette lines will be displayed for a surface type if it has silhouette lines turned on (using an option such as plane_silhouettes, cone_silhouettes, etc.) and this option is also on. If this option is off, silhouettes are not displayed for any surface type, even if individual surface types have silhouette lines turned on. This option only applies to the Test Harness.

Example: Not applicable

sketch_mesh

Option: Modeler Control

Action: Sets which mesh elements are sketched in the Test Harness.

Name String: **sketch_mesh**

Scheme: Not applicable

Test Harness: string “none”, “all”, “face” “face”

C++: Not applicable

Description: When set to “none”, no elements are drawn. When set to “all”, all elements on the surface are sketched. When set to “face”, only face interior elements and boundary elements are sketched. This option only applies to the Test Harness.

Example: Not applicable

sphere_param_lines

Option: Modeler Control

Action: Enables or disables the display of parameter lines on spheres in the Test Harness.

Name String: **sphere_param_lines**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: Turning on this option enables the display of sphere parameter lines. The number of parameter lines is set using the options u_param_lines and v_param_lines. This option only applies to the Test Harness.

Example: Not applicable

sphere_silhouettes

Option: Modeler Control

Action: Enables of disables display of a sphere silhouette in the Test Harness.

Name String: **sphere_silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: If on, enables display of a sphere silhouette. Option silhouettes must also be on. This option only applies to the Test Harness.

Example: Not applicable

spline_param_lines

Option: Modeler Control

Action: Enables or disables the display of parameter lines on splines in the Test Harness.

Name String: **spline_param_lines**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: Turning on this option enables the display of spline parameter lines. The number of parameter lines is set using the options u_param_lines and v_param_lines. This option only applies to the Test Harness.

Example: Not applicable

spline_silhouettes

Option: Modeler Control

Action: Enables or disables display of a spline silhouette in the Test Harness.

Name String: **spline_silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: If on, enables display of a spline silhouette. Option silhouettes must also be on. This option only applies to the Test Harness.

Example: Not applicable

torus_param_lines

Option: Modeler Control

Action: Enables or disables the display of parameter lines on tori in the Test Harness.

Name String: **torus_param_lines**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: When on, enables the display of parameter lines on tori. The number of parameter lines is set using the options u_param_lines and v_param_lines. This option only applies to the Test Harness.

Example: Not applicable

torus_silhouettes

Option: Modeler Control

Action: Enables or disables the display of torus silhouettes in the Test Harness.

Name String: **torus_silhouettes**

Scheme: Not applicable

Test Harness: integer 0, 1 1

C++: Not applicable

Description: When on, enables the display of torus silhouettes. Option silhouettes must also be on. This option only applies to the Test Harness.

Example: Not applicable

u_param_lines

Option: Modeler Control

Action: Sets the number of *u* parameter lines to display in the Test Harness.

Name String: **u_param_lines**

Scheme: Not applicable

Test Harness: integer 0

C++: Not applicable

Description: This is used to calculate equally spaced *u* isoparametric curves on a face, based on the number of lines specified. If the number is less than or equal to zero, no parameter lines will be displayed. This option only applies to the Test Harness.

Example: Not applicable

v_param_lines

Option: Modeler Control

Action: Sets the number of *v* parameter lines to display in the Test Harness.

Name String: **v_param_lines**

Scheme: Not applicable

Test Harness: integer 0

C++: Not applicable

Description: This is used to calculate equally spaced *v* isoparametric curves on a face, based on the number of lines specified. If the number is less than or equal to zero, no parameter lines will be displayed. This option only applies to the Test Harness.

Example: Not applicable