

# Chapter 7.

## 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.

### full\_error\_msg

Option: Error Handling, Modeler Control

Action: Controls the amount of information issues for ACIS error messages.

Name String: **full\_error\_msg**

Scheme: boolean      #t, #f      #f

Test Harness: Not applicable

C++: logical      TRUE, FALSE      FALSE

Description: This option controls whether ACIS errors get reported in full, including the error module and identifier (TRUE), or merely with the text message (FALSE, default).

Example: 

```
; full_error_msg
; Print error module and identifier with error
; messages
(option:set "full_error_msg" #t)
;; #f
```

### journal\_result\_limit

Option: Scheme Interface, Modeler Control

Action: Sets the maximum number of characters used to journal the result of a Scheme operation when journaling is turned on.

Name String: **journal\_result\_limit**

Scheme: integer >0 50

Test Harness: Not applicable

C++: Not applicable

Description: Refer to Action.

Example: 

```
; journal_result_limit
; Set larger limit
(option:set "journal_result_limit" 100)
;; 50
```

## save\_version

Option: Scheme Interface, SAT Save and Restore, Modeler Control

Action: Sets the ACIS version to be used by the part:save Scheme extension.

Name String: **save\_version**

Scheme: integer 3–digit ACIS version installed version

Test Harness: Not applicable

C++: Not applicable

Description: The ACIS version controls the save file format. The first digit is the major version number. The second and third digits are the minor version number. For example, version 2.1 is represented as 201, and version 3.0 is represented as 300. The default value is the currently installed version of ACIS. In C++ applications, the API `api_save_version` is used to set the version.

Example:

```
; save_version
; Create a solid block.
(solid:block (position 0 0 0) (position 10 10 10))
;; #[entity 2 1]
; Save the currently-active part to the named file.
(part:save "test")
;; #t
; Set the save_version option to ACIS 5.0.
(option:set "save_version" 500)
;; 700
; Save the solid block again in ACIS 5.0 format.
(part:save "test50")
;; #t
; Reset the save_version option to ACIS 6.0.
(option:set "save_version" 600)
;; 500
```