

Appendix B.

Syntax Summary

Topic:

Ignore

This appendix provides a syntax summary of reference items defined in the . There is a separate section for each item type (Scheme extension, test harness command, etc.), and the items within each section are alphabetized. Just the syntax of each item is given; refer to the item's reference template for a complete description.

Functions

Topic:

Ignore

```
PART_CONTEXT* active_part_context()

outcome api_pm_add_entity(ENTITY* entity, PART* part)

outcome api_pm_create_part(unsigned int, PART*& part)

outcome api_pm_delete_all_states(HISTORY_STREAM* hs = NULL)

outcome api_pm_delete_part(PART* part)

outcome api_pm_entity_id(ENTITY* ent, entity_id_t& id, PART*&
    part)

outcome api_pm_load_part(FILE* fp, logical text_mode, PART*
    the_part, logical with_history, ENTITY_LIST& new_entities)

outcome api_pm_lookup_entity(entity_id_t id, PART* part, ENTITY*&
    ent)

outcome api_pm_name_state(const char* name, HISTORY_STREAM* hs =
    NULL)

outcome api_pm_note_state(outcome out, int& depth)

outcome api_pm_part_entities(PART* part, entity_filter* filter,
    ENTITY_LIST& ent)
```

```

outcome api_pm_remove_entity(ENTITY* entity)

outcome api_pm_roll_n_states(int n_wanted, HISTORY_STREAM* hs,
    int& n_actual)

outcome api_pm_roll_to_state(const char* name, HISTORY_STREAM*
    hs, int& n_actual)

outcome api_pm_save_part(FILE* fp, logical text_mode, PART*
    the_part, logical with_history = 0, logical mainline_only = 0)

outcome api_pm_start_state(int& depth)

void delete_GC_Objects()

PART_CONTEXT* get_part_context(const ENTITY* ent)

scheme_error_callback_list& get_scheme_error_callback_list()

const char* get_Scm_String(ScmObject s)

logical is_Scm_Real_List(ScmObject list)

void refresh_all()

int SchemeEvaluate(ScmObject expr, ScmObject& result)

int SchemeEvaluate(const char* str)

int SchemeEvaluate(const char* str, ScmObject& result)

int SchemeEvaluate(const char* str, param_string& result_string)

int SchemeLoad(const char* filename)

int scheme_process(const char* inpLine, int echo)

void start_entity_creation()

void start_entity_creation(void)

void start_entity_modification()

void start_entity_modification(void)

```

Scheme Extensions

Topic: [Ignore](#)
 (check-interrupt)

```
(dl-item:color item)
(dl-item:display item)
(dl-item:erase item)
(dl-item:point position)
(dl-item:polyline position-list [fill=#f])
(dl-item:set-color item color)
(dl-item:text position string)
(dl-item? object)
(edge:chain start-edge-entity [stop-edge-entity])
(entity:color entity)
(entity:display entity-list)
(entity:displayed? entity)
(entity:erase entity-list)
(entity:highlighted? entity)
(entity:part entity)
(entity:select entity-list {on-off})
(entity:selected? entity)
(entity:set-color entity-list color)
(entity:set-highlight entity-list {on-off})
(entity:set-part entity [part=active])
(entity:set-render entity-list on-off=#t)
(entray:position entray)
(entray:vertex entray)
(env:active-part)
(env:active-view)
(env:active-wcs-color)
```

```
(env:auto-display)
(env:count-parts)
(env:default-color)
(env:highlight-color)
(env:parts)
(env:point-size)
(env:point-style)
(env:set-active-part part)
(env:set-active-view view)
(env:set-active-wcs-color color=0 1 0)
(env:set-auto-display on-off=#t)
(env:set-default-color color=0 1 0)
(env:set-highlight-color color=1 1 1)
(env:set-point-size size=10)
(env:set-point-style style-string=x)
(event x y button view [state])
(event:alt? event)
(event:button event)
(event:control? event)
(event:left? event)
(event:middle? event)
(event:ray event)
(event:right? event)
(event:shift? event)
(event:view event)
(event:x event)
```

```
(event:y event)
(event? object)
(filter:color color)
(filter:display)
(gvector:view x y z [view=active])
(history part | entity | history-id
 [part-id] | "default")
(history:get-modified-faces mod-type-string
 [delta-state-name-string=null] [history])
(history:load filename [textmode=#t] [part=active])
(history:save filename [text-mode=#t]
 [save-type="entire"] [history])
(history:size hist)
(journal:abort)
(journal:append [filename])
(journal:load filename)
(journal:off)
(journal:on [filename])
(journal:pause)
(journal:resume)
(journal:save filename)
(journal:step value)
(load-dll filename)
(part id)
(part:clear [part=active])
(part:clear-selection [part=active])
(part:close [part=active])
```

```
(part:debug [part=active])
(part:delete-rendering-context [part=active] name)
(part:entities [part=active] [filter=NULL])
(part:get-distribution-mode)
(part:load filename [textmode=#t] [part=active]
  [with-history=#f])
(part:modified? [part=active])
(part:name [part=active])
(part:new [size=1009])
(part:save [filename=partname.sat] [textmode=#t]
  [part=active] [with-history=#f]
  [mainline-only=#f])
(part:save-selection ent-list filename [textmode=#f]
  [with-history=#f])
(part:selection [part=active])
(part:set-distribution-mode on-off)
(part:set-name [name=id name] [part=active])
(part:views [part=active])
(part? object)
(pick:aperture)
(pick:edge {event | pick-ray} [filter])
(pick:entity event [filter] [depth])
(pick:face {event | pick-ray} [face-index] [filter])
(pick:in-region event1 event2 [filter])
(pick:position event [plane-position
  [plane-direction=xy-plane]])
(pick:ray event [plane-position
  [plane-direction=xy-plane]])
(pick:set-aperture dx=15 dy=15)
```

```
(pick:vertex {event | pick-ray} [filter])
(position:view x y z [view=active])
(read-event)
(ro:color item)
(ro:delete rendering-object)
(ro:display rendering-object)
(ro:erase rendering-object)
(ro:line pos1 pos2)
(ro:new view-opt)
(ro:point position)
(ro:polyline pos-list [fill=#f])
(ro:text position text)
(roll [number-of-states=-1] [name-string=NULL]
  [end] [start] [history])
(roll:back? [history])
(roll:debug history [level] [ent-name])
(roll:delete-all-states [history])
(roll:delete-following-states [history])
(roll:delete-inactive-states [history])
(roll:delete-previous-states [history])
(roll:forward? [history])
(roll:get-logging)
(roll:mark-end)
(roll:mark-start)
(roll:merge-delta-state name [stream])
(roll:merge-delta-states [id1 | name1] [id2 | name2]
  [stream] [prune-partners=#f])
```

```
(roll:name-state name-string [history])
(roll:named-states name-pattern [history])
(roll:set-logging on-off)
(roll:set-max-states num [history])
(roll:to-state-id state-id [history])
(system:bell [duration=hardware-settings]
  [frequency=hardware-settings])
(system:command cmd)
(system:getenv name-string)
(system:name)
(system:play-sound filename [async=1])
(system:set-timer-off)
(system:set-timer-on num-sec)
(system:sleep duration)
(system:time-string)
(ui:error-dialog msg-string [x y
  width=string-dependent height=107])
(ui:info-dialog msg-string [x y
  width=string-dependent height=111])
(ui:prompt msg-string)
(ui:warning-dialog msg-string [x y
  width=string-dependent height=109])
(ui:yesno-dialog msg-string [title-string=ACIS]
  [x y width=string-dependent height=109])
(versionid)
(view:bg-color [view=active])
(view:clear [view=active])
(view:delete [view=active])
```

```
(view:display-facets disp)
(view:display-param-lines disp)
(view:display-surface-polys disp)
(view:draw-point position [view=active])
(view:draw-polyline position-list [fill]
  [view=active])
(view:draw-text position string [view=active])
(view:eye [view=active])
(view:fg-color [view=active])
(view:handle [view=active])
(view:height [view=active])
(view:hither [view=active])
(view:out [view=active])
(view:perspective? [view=active])
(view:right [view=active])
(view:set eye-position target-position
  up-direction [view=active])
(view:set-bg-color color [view=active])
(view:set-clipping hither=0.025 yon [view=active])
(view:set-draw-mode mode [view=active])
(view:set-eye eye-position=(0,0,0) [view=active])
(view:set-fg-color color [view=active])
(view:set-font font-string [view=active])
(view:set-font-size size [view=active])
(view:set-hither hither=0.025 [view=active])
(view:set-line-style style-string [view=active])
(view:set-line-width size=1 [view=active])
```

```
(view:set-perspective persp-orth [view=active])
(view:set-point-size size=10 [view=active])
(view:set-point-style style-string [view=active])
(view:set-rb-mode on-off [view=active])
(view:set-size width height [view=active])
(view:set-target target-position=(0,0,0)
  [view=active])
(view:set-title name-string [view=active])
(view:set-up up-direction=(0,0,1) [view=active])
(view:set-viewport [orgx orgy width height]
  [view=active])
(view:set-yon yon [view=active])
(view:target [view=active])
(view:up [view=active])
(view:viewport [view=active])
(view:width [view=active])
(view:with-handle handle)
(view:yon [view=active])
(view? object)
(void? object)
```