

Appendix B.

Syntax Summary

Topic: Ignore

This appendix provides a syntax summary of reference items defined in the Component Name. 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

```
outcome api_check_ct(const ENTITY* ent, insanity_list*& list,
    AcisOptions* ao = NULL)

outcome api_ct_add_to_group(ENTITY* ent, SPAGROUP* group,
    AcisOptions* ao = NULL)

outcome api_ct_attach(ENTITY_LIST& body_list, AcisOptions* ao =
    NULL)

outcome api_ct_attach_cells(LUMP* lump, AcisOptions* ao = NULL)

outcome api_ct_cell_area(CELL* cell, double req_rel_accy, double&
    area, double& est_rel_accy_achieved, AcisOptions* ao = NULL)

outcome api_ct_cell_mass_pr(CELL3D* cell, SPAposition const&
    root_proj_pl, SPAunit_vector const& normal_proj_pl, int
    selector, double req_rel_accy, double& volume, SPAposition&
    cofg, tensor& inertia, double p_moments[], SPAunit_vector*
    p_axes[], double& est_rel_accy_achieved, AcisOptions* ao =
    NULL)

outcome api_ct_copy_cell(CELL* cell, BODY*& body, AcisOptions* ao
    = NULL)

outcome api_ct_expand(ENTITY_LIST& body_list, AcisOptions* ao =
    NULL)
```

```

outcome api_ct_expand_cells(LUMP* lump, AcisOptions* ao = NULL)
outcome api_ct_flatten(ENTITY_LIST& body_list, AcisOptions* ao =
    NULL)
outcome api_ct_flatten_cells(LUMP* lump, AcisOptions* ao = NULL)
outcome api_ct_get_all_cells(ENTITY_LIST const& body_list,
    ENTITY_LIST& cell_list, AcisOptions* ao = NULL)
outcome api_ct_lose_cells(LUMP* lump, AcisOptions* ao = NULL)
outcome api_ct_lose_group(SPAGROUP* group, AcisOptions* ao =
    NULL)
outcome api_ct_make_group(ENTITY_LIST& list, SPAGROUP*& group,
    AcisOptions* ao = NULL)
outcome api_ct_point_in_cell(SPAposition const& test_point,
    CELL3D* target_cell, point_containment& pc, AcisOptions* ao =
    NULL)
outcome api_ct_propagate_cface_attribs(LUMP* lump, AcisOptions*
    ao = NULL)
outcome api_ct_remove(ENTITY_LIST& body_list, AcisOptions* ao =
    NULL)
outcome api_ct_remove_from_group(ENTITY* ent, SPAGROUP* group,
    AcisOptions* ao = NULL)
outcome api_ct_return_ents(SPAGROUP* group, ENTITY_LIST& list,
    AcisOptions* ao = NULL)
outcome api_ct_return_groups(ENTITY* ent, ENTITY_LIST& list,
    AcisOptions* ao = NULL)
outcome api_ct_vacate_cell(CELL3D* cell, AcisOptions* ao = NULL)
outcome api_get_edges_from_all_entities(ENTITY* ent, ENTITY_LIST&
    edge_list, PAT_NEXT_TYPE include_pat = PAT_CAN_CREATE,
    AcisOptions* ao = NULL)
outcome api_get_faces_from_all_entities(ENTITY* ent, ENTITY_LIST&
    face_list, PAT_NEXT_TYPE include_pat = PAT_CAN_CREATE,
    AcisOptions* ao = NULL)
outcome api_get_vertices_from_all_entities(ENTITY* ent,
    ENTITY_LIST& vertex_list, PAT_NEXT_TYPE include_pat =
    PAT_CAN_CREATE, AcisOptions* ao = NULL)

```

```

outcome api_initialize_cellular_topology()
outcome api_terminate_cellular_topology()
logical is_ATTRIB_CELL(const ENTITY* e)
logical is_ATTRIB_CFACE_VOL(const ENTITY* e)
logical is_ATTRIB_CT(const ENTITY* e)
logical is_ATTRIB_FACECFACE(const ENTITY* e)
logical is_ATTRIB_VOL_COL(const ENTITY* e)
logical is_CELL(const ENTITY* e)
logical is_CELL2D(const ENTITY* e)
logical is_CELL3D(const ENTITY* e)
logical is_CFACE(const ENTITY* e)
logical is_CSHELL(const ENTITY* e)
logical is_SPAGROUP(const ENTITY* e)

```

Scheme Extensions

```

Topic: Ignore
(cell:2d? cell)
(cell:3d? cell)
(cell:area cell [accuracy=0.1])
(cell:attach entity-list)
(cell:classify-position cell position)
(cell:copy cell)
(cell:expand entity-list)
(cell:find face1 [face2])
(cell:flatten entity-list)
(cell:massprop cell [selector accuracy
proj-root proj-normal])

```

```
(cell:remove entity-list)
(cell? entity)
(entity:cells body-list)
(entity:edges entity-list)
(entity:faces entity-list)
(entity:vertices entity-list)
```