

Appendix A.

Reference Summary

Topic: Ignore

This appendix provides a summary of the reference items defined in the Component Name. Just the name and a brief description of each item is given. There is a separate section for each type of item (class, function, Scheme extension, test harness command, etc.).

Classes

Topic: Ignore

pcurve_interp Fits a parameter space curve.

PRIMITIVE_ANNOTATION Used for primitive annotations.

splgrid Defines a spline surface grid.

splsurf Defines a spline surface.

Functions

Topic: Ignore

api_accurate_bs3_approximation Recomputes the 3D B-spline surface approximation, using the newer algorithm.

api_body Creates an empty body.

api_body_mass_pr Determines the mass properties of a body.

api_body_to_2d Converts single-sided faces to double-sided faces.

api_build_wire Creates a wire from an array of positions and an array of pointers to curves.

api_closed_wire Determines if a wire or a single-wire body is closed.

`api_create_point` Creates a point entity at the specified position.
`api_create_text` Creates a text entity at given position.
`api_curve_arc` Creates a circular arc given the center, radius, and start and end angles.
`api_curve_arc_3curve` Creates a circle tangent to three curves.
`api_curve_arc_3pt` Creates a circle or circular arc through three positions.
`api_curve_arc_center_edge` Creates a circle or circular arc given a center and two edge points.
`api_curve_arc_diagonal` Creates a circle or circular arc given two points on the diagonal.
`api_curve_bezier` Creates a cubic Bezier curve given four control points.
`api_curve_ellipse` Creates an ellipse parallel to the xy-plane of the active working coordinate system.
`api_curve_fillet` Creates a fillet between two curves.
`api_curve_law` Creates a curve from a law.
`api_curve_line` Creates a line through two positions.
`api_curve_line_tangent` Creates a line tangent to two curves or through a position and tangent to a curve.
`api_curve_spline` Creates a spline curve that interpolates an array of positions.
`api_curve_spline2` Creates a spline curve that interpolates an array of positions at specified parameter values.
`api_edge` Creates a new edge which is a copy of the specified edge.
`api_edge_arclength_metric` Creates a arclength parameterized curve from a curve.
`api_edge_arclength_param` Creates a new arcwise edge which is a copy of the specified edge.
`api_edge_law` Creates an edge from a law.

`api_edge_plaw` Creates an edge from a R2 law and a face.
`api_edge_spiral` Creates an edge from a spiral definition.
`api_edge_spring` Creates an edge from a spring definition.
`api_edge_spring_law` Creates an edge from a spring definition and a radius law.
`api_edge_spring_taper` Creates an edge from a tapered spring definition.
`api_edge_to_spline` Creates a spline edge using a given edge.
`api_ed_inters_to_ents` Creates vertex and edge entities corresponding to a list of edge-edge intersections as produced by `api_inter_ed_ed`.
`api_enclose_void` Modifies an enclosing set of faces from void bounding to material bounding.
`api_ent_area` Determines the area of a face, shell, lump, or body.
`api_face_conic` Creates a conical face.
`api_face_cylinder_cone` Creates a cylindrical or conical face.
`api_face_law` Creates a law face.
`api_face_plane` Creates a planar face.
`api_face_sphere` Creates a spherical face.
`api_face_spl_apprx` Creates a spline face which is fit to a grid of positions.
`api_face_spl_ctrlpts` Creates a spline face from control points and knots.
`api_face_spl_intrp` Creates a spline surface face.
`api_face_torus` Creates a toroidal face.
`api_fillet_vertex` Fillets a wire at a given vertex.
`api_find_face` Determines a planar face with normal in given direction.

api_find_vertex Gets the vertex closest to given position.
 api_initialize_constructors Initializes the constructor library.
 api_loop_external Determines if a loop is internal or external.
 api_make_approx_curve Creates a curve that approximates a given curve.
 api_make_approx_surface Creates a surface that approximates a given surface.
 api_make_cnface Creates a face that is a portion of a cone.
 api_make_cuboid Creates cuboid of given width, depth and height.
 api_make_edge_from_curve Creates an edge using a copy of the input curve.
 api_make_ewire Creates a body that consists a single wire from a list of edges that are connected in the order given with no branching.
 api_make_ewires Separates and sorts a possibly branched list of edges into a group of non-branched wire bodies.
 api_make_frustum Creates an elliptical cone or cylinder of given height and radii.
 api_make_kwire Creates a kwire body from an array of positions and bulges.
 api_make_planar_disk Creates a face that is a planar disk.
 api_make_plface Creates a face that is a parallelogram specified by three points: origin, left, and right.
 api_make_polygon Creates a polygonal wire from in a plane orthogonal to a given normal.
 api_make_prism Creates an elliptical prism of given height, radii, and number of sides.
 api_make_pyramid Creates an elliptical pyramid of given height, radii, and number of sides.
 api_make_spface Creates a face that is a portion of a sphere.
 api_make_sphere Creates a sphere of given radius.

<code>api_make_spline</code>	Creates a body consisting of a single face which is the whole of a given B-spline surface.
<code>api_make_torus</code>	Creates a torus of given major and minor radii.
<code>api_make_trface</code>	Creates a face that is a portion of a torus.
<code>api_make_wire</code>	Creates a polygonal wire from an array of positions.
<code>api_manifold_class</code>	Determines the differences between all manifold shells, manifold sheets, and wire edges. Returns all nonmanifold edges and vertices.
<code>api_mk_by_faces</code>	Adds each FACE in the array that is attached to a SHELL within a LUMP to a BODY.
<code>api_mk_ed_bs3_curve</code>	Creates an EDGE that represents an intersection curve defined by a 3D NURBS curve.
<code>api_mk_ed_conic</code>	Creates a spline edge defined by a conic.
<code>api_mk_ed_cubic</code>	Creates an EDGE that represents a cubic spline curve interpolating or fitting a sequence of points.
<code>api_mk_ed_ellipse</code>	Creates an EDGE that represents a bounded elliptical arc.
<code>api_mk_ed_int_ctrlpts</code>	Creates an EDGE that represents an intersection curve defined by a sequence of control points and knots.
<code>api_mk_ed_line</code>	Creates an EDGE that represents a bounded line segment between two points.
<code>api_mk_fa_spl_ctrlpts</code>	Creates a FACE representing the spline surface defined by a sequence of control points and knots.
<code>api_mk_fa_spl_fit</code>	Creates a FACE that represents a spline surface defined by a sequence of points on the surface.
<code>api_mk_fa_spl_intp</code>	Creates a FACE that represents a spline surface defined by a sequence of points on a surface.
<code>api_modify_ellipse</code>	Modifies an argument of an ellipse.

api_modify_line Modifies line end points.
 api_planar_face_pr Locates area, center of area, second moments,
 and principal axes of a planar face.
 api_q_edges_around_vertex Gets a list of edges that share a given vertex.
 api_reverse_body Reverses the orientations of all coedges in the
 body. Also reverses face orientations and the
 orientation of loops within the faces.
 api_reverse_face Reverses the sense of a face.
 api_reverse_wire Reverses the direction (sense) of a wire.
 api_sheet_from_ff Creates a sheet body as a copy of a face.
 api_shell_external Determines whether a shell is internal or
 external.
 api_solid_block Creates a solid block given two diagonal
 positions.
 api_solid_cylinder_cone Creates a solid cylinder or cone given two
 positions on the axis.
 api_solid_sphere Creates a solid sphere given the center and
 radius.
 api_solid_torus Creates a solid torus given the center and major
 and minor radii.
 api_split_curve Splits a curve at a position or its intersection
 with another curve.
 api_split_edge_at_disc Splits a curve at G1 or G2 discontinuities.
 api_terminate_constructors Terminates the constructor library.
 api_trans_edge Creates a new edge which is a transformed copy
 of the specified edge.
 api_trim_2curves Trims two curves to their intersection.
 api_trim_chain Trims a list of curves so that they form a
 contiguous chain.
 api_trim_curve Trims one end of a curve at a position or its
 intersection with another curve.

api_trim_middle	Trims the middle of a curve.
api_wiggle	Creates a “wiggle” body for testing purposes.
api_wire_len	Determines the length of a wire or a wire body.
api_wire_to_chain	Gets a list of the coedges of a body that is a single unbranched wire.
is_PRIMITIVE_ANNOTATION	Determines if an ENTITY is a PRIMITIVE_ANNOTATION.
make_face_spline	Constructs a new FACE from a spline.

Options

Topic:	Ignore	
props		Sets insertion of a property in an elliptical cylinder or frustum (cone).
spl_edges_forward		Controls the direction of the underlying curve and the edge sense when a spline edge is created from an input edge.

Scheme Extensions

Topic:	Ignore	
arc:set-center		Sets the center position of a circular curve or edge or an elliptical curve or edge.
arc:set-direction		Sets the major axis direction of an elliptical curve or edge.
arc:set-end		Sets the end angle of a circular curve or edge or an elliptical curve or edge.
arc:set-normal		Sets the normal of a circular curve or edge or an elliptical curve or edge.
arc:set-radius		Sets the radius of a circular curve or edge or an elliptical curve or edge.
arc:set-radius-ratio		Sets the radius ratio of an elliptical curve or edge.



<code>arc:set-start</code>	Sets the start angle of a circular curve or edge or an elliptical curve or edge.
<code>body:find-face</code>	Returns a face of the given body.
<code>body:reverse</code>	Reverses the sense on all faces, lumps, wire coedges, and wires within a body.
<code>coedge:add-pcurve</code>	Adds a pcurve to a coedge.
<code>curve:circular</code>	Creates a temporary circular curve.
<code>curve:elliptical</code>	Creates a temporary elliptical curve.
<code>curve:linear</code>	Creates a temporary linear-curve.
<code>edge:arclength-metric</code>	Calculates an arclength metric for an edge.
<code>edge:arclength-param</code>	Creates an arclength parameterized edge from an edge.
<code>edge:bezier</code>	Creates a cubic bezier curved edge from four control points.
<code>edge:bezier-ndeg</code>	Creates a bezier curved edge of arbitrary degree (<code>ctrlpts-1</code>) from a list of control points.
<code>edge:circular</code>	Creates an arc with the specified center position and radius.
<code>edge:circular-3curve</code>	Creates an edge specified by an <code>edge:circular</code> tangent to three curves.
<code>edge:circular-3pt</code>	Creates an arc passing through three positions.
<code>edge:circular-center-rim</code>	Creates an arc given a center position and one or two positions on the arc.
<code>edge:circular-diameter</code>	Creates an arc passing through two positions based on the diameter.
<code>edge:conic</code>	Creates a rho conic edge in which the geometrical definition represents a hyperbola or parabola.
<code>edge:ellipse</code>	Creates an edge that is part of an ellipse with a given center, normal, major axis, and radius ratio.

edge:elliptical Creates a full or partial ellipse by specifying the start and end angles.

edge:end Returns the ending position of the edge.

edge:end-dir Returns the end direction of the curve.

edge:extend Creates a new edge that is an extension of an existing edge.

edge:fillet Creates a fillet blend on two edge entrays (entities with rays).

edge:from-curve Creates an edge from a curve.

edge:get-tolerance Gets the tolerance of a TEDGE.

edge:law Creates an edge from a law.

edge:length Returns the length of the edge.

edge:linear Creates a linear-edge between two locations.

edge:mid-point Returns the middle point position of an edge.

edge:mid-point-dir Returns the direction of the midpoint of the curve.

edge:plaw Creates an edge from a law defining a parameter space curve on a surface.

edge:reverse Reverses the sense of an edge.

edge:spiral Creates an edge from a spiral definition.

edge:spline Creates a continuous spline edge from a list of positions.

edge:spline2 Creates a continuous spline edge from a list of positions and parameter values.

edge:split Splits an edge into two entities at a specified position.

edge:split-at-disc Splits an edge up to G1 or G2 discontinuities.

edge:spring Creates an edge from a spring definition.

edge:spring-law Creates an edge from a spring definition and a radius law.

edge:spring-taper Creates an edge from a tapered spring definition.

edge:start Returns the starting position of the edge.

edge:start-dir Returns the start direction of the curve.

edge:tolerant Replaces an edge with a tolerant edge.

edge:trim Trims an entray to the bounds of the specified position or another edge (entry).

edge:trim-chain Trims a list of edges to form a continuous chain with each pair of edges having a common end point.

edge:trim-intersect Trims both edges to the intersection of the edges.

edge:trim-middle Trims the middle out of an edge.

entity:accurate-approx Regenerates bs3 surfaces, using a slower but more robust approach.

entity:box Gets an entity's extrema box.

entity:tolerize Finds all the edges and vertices in a part that are not accurate to within a given tolerance, and turns them into tolerant entities.

face:area Gets the area of a face.

face:cone Creates a conical face relative to the active WCS.

face:conic Creates a conic surface with a given radius R, a conic constant K, and an extent E such that $x^2+y^2 < E$.

face:cylinder Creates a cylindrical face relative to the active WCS.

face:extend Creates a new face that is an extension of an existing face.

face:law Creates a law face.

face:par-box Gets the parameter ranges of a face.

face:planar-disk Creates a planar disk of a given center, normal, and radius.

face:plane Creates a face that is a parallelogram specified by three points: origin, left, and right.

face:prop Analyzes the properties of a face.

face:reverse Reverses the sense of a face.

face:sphere Creates a spherical face.

face:spline-ctrlpts Creates a spline face using control points.

face:spline-grid Creates a spline face using a grid.

face:torus Creates a toroidal face.

line:set-direction Sets the direction of a linear-curve or edge.

line:set-end Sets the end position of a linear-curve or edge.

line:set-start Sets the start position of a linear-curve or edge.

loop:external? Determines if a loop is internal or external.

point Creates a point at the specified position.

sheet:1d Modifies a double-sided sheet body into a single-sided sheet body.

sheet:2d Modifies a single-sided sheet body into a double-sided sheet body.

sheet:enclose Converts a closed 2D sheet into a solid volume.

sheet:face Creates a sheet body from a given face.

solid:area Gets the surface area of a solid.

solid:block Creates a solid block.

solid:closed? Verifies if a Scheme object is a closed solid.

solid:cone Creates a right circular cone.

solid:cylinder Creates a right circular cylinder.

solid:manifold? Verifies if a Scheme object is a manifold solid.

<code>solid:massprop</code>	Analyzes the mass properties of a solid.
<code>solid:prism</code>	Creates a solid prism.
<code>solid:pyramid</code>	Creates a solid pyramid.
<code>solid:sphere</code>	Creates a sphere centered at the specified position.
<code>solid:torus</code>	Creates a solid torus.
<code>solid:wiggle</code>	Creates a rectangular block with a spline surface top.
<code>splgrid</code>	Creates a spline grid structure data type.
<code>splgrid:copy</code>	Copies a spline grid structure data type.
<code>splgrid:point-item</code>	Gets a single point from a list of points of a spline grid data structure.
<code>splgrid:print</code>	Prints the <code>splgrid</code> data.
<code>splgrid:set-point-item</code>	Modifies a single point in a list of points of a spline grid.
<code>splgrid:set-point-list</code>	Sets a list of points that define a spline grid into a spline grid data structure.
<code>splgrid:set-tolerance</code>	Sets the point tolerance that is used when approximating a spline surface from a grid of points.
<code>splgrid:set-u-tanvec-item</code>	Sets a single vector in the starting or ending tangent vector list, in u , of a spline surface.
<code>splgrid:set-u-tanvec-list</code>	Sets the tangent vector list for the starting or ending boundary conditions, in u , of a spline surface.
<code>splgrid:set-v-tanvec-item</code>	Sets a single vector in the starting or ending tangent vector list, in v , of a spline surface.
<code>splgrid:set-v-tanvec-list</code>	Sets the tangent vector list for the starting or ending boundary conditions, in v , of a spline surface.
<code>splgrid:tolerance</code>	Gets the point tolerance that is used when approximating a spline surface from a grid of points.

<code>splgrid:u-points</code>	Gets the number of points (columns) in the u -direction of a spline surface grid.
<code>splgrid:u-tanvec-item</code>	Gets a single vector from the starting or ending tangent vector list, in u , of a spline surface.
<code>splgrid:v-points</code>	Gets the number of points (rows) in the v -direction of a spline surface grid.
<code>splgrid:v-tanvec-item</code>	Gets a single vector from the starting or ending tangent vector list, in v , of a spline surface.
<code>splgrid?</code>	Determines if a Scheme object is of the type spline grid data structure.
<code>splsurf</code>	Creates a spline surface data type.
<code>splsurf:copy</code>	Copies a spline surface data type.
<code>splsurf:ctrlpt-count</code>	Gets the number of control points from a spline surface.
<code>splsurf:ctrlpt-item</code>	Gets a single control point from a list of points of a spline surface.
<code>splsurf:param</code>	Gets a u or v parameter value for a spline surface.
<code>splsurf:print</code>	Prints spline surface data.
<code>splsurf:set-ctrlpt-item</code>	Modifies a single control point in a list of points of a spline surface.
<code>splsurf:set-ctrlpt-list</code>	Sets a list of control points for a spline surface into a <code>splsurf</code> data structure.
<code>splsurf:set-u-knot-item</code>	Modifies a single knot value, in u , for a spline surface.
<code>splsurf:set-u-knot-list</code>	Sets a list of knots, in u , for a spline surface.
<code>splsurf:set-u-param</code>	Modifies a u parameter value for a spline surface.
<code>splsurf:set-v-knot-item</code>	Modifies a single knot value, in v , for a spline surface.
<code>splsurf:set-v-knot-list</code>	Sets a list of knots, in v , for a spline surface.

<code>splsurf:set-v-param</code>	Modifies a <i>v</i> parameter value for a spline surface.
<code>splsurf:set-weight-item</code>	Modifies a single weight in a list of weights for the control points of a rational spline surface.
<code>splsurf:set-weight-list</code>	Sets a list of weights for the control points of a rational spline surface.
<code>splsurf:u-knot-count</code>	Gets the number of knots, in <i>u</i> , for a spline surface.
<code>splsurf:u-knot-item</code>	Gets a single knot value, in <i>u</i> , for a spline surface.
<code>splsurf:v-knot-count</code>	Gets the number of knots, in <i>v</i> , for a spline surface.
<code>splsurf:v-knot-item</code>	Gets a single knot value, in <i>v</i> , for a spline surface.
<code>splsurf:weight-item</code>	Gets a single weight value from a list of weights for the control points of a rational spline surface.
<code>splsurf?</code>	Determines if a Scheme object is a spline surface data structure.
<code>text</code>	Creates a text entity.
<code>tolerant:fix</code>	Modifies all of the edges or coedges to make them tolerant.
<code>tolerant:move</code>	Moves geometry in the direction specified.
<code>tolerant:none</code>	Modifies all tedges or tcoedges to make them exact.
<code>tolerant:optimize</code>	Minimize all TVERTEX tolerant on edges.
<code>tolerant:optimized?</code>	Checks to see if the tolerance on an edge optimized.
<code>tolerant:report</code>	Returns the vertex and edge with the largest tolerances on the specified entity.
<code>tolerant:update</code>	Updates all tedges or tcoedges tolerance.

tolerant? Checks to see if an entity is tolerant or has tolerant topology.

vertex:fillet Creates a fillet blend on two vertices.

vertex:get-tolerance Returns the tolerance on a TVERTEX.

vertex:tolerant Replaces a vertex with a tolerant vertex.

wire-body Creates a wire body from a list of edges.

wire-body:kwire Creates a planar wire specified by a sequence of points and “bulge” factors.

wire-body:length Gets the length of a wire.

wire-body:points Creates a wire body from a list of positions.

wire-body:polygon Creates a regular polygon wire-body.

wire-body:unbranch Splits a branched wire body into its component branches.

wire:end Gets the ending position of a wire.

wire:reverse Reverses the sense of a wire.

wire:start Gets the starting position of a wire.