

Appendix A.

Reference Summary

Introduction

This appendix provides a summary of the reference items defined in ACIS. 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.). Within each section, the items are organized by the primary mechanism(s) in which they participate. An item may appear in more than one mechanism.

Enumerations

AF_ADJUST_MODE	Capitalized text used to specify adjustment mode.
AF_GRID_MODE	Capitalized text used to specify grid mode.
AF_SURF_MODE	Capitalized text used to specify surface mode.
AF_TRIANG_MODE	Capitalized text used to specify triangulation mode.
CURVE_EXTENSION_TYPE	Number specifies the extension type.
bl_continuity	Text specifies blending continuity.
bl_convexity	Text specifies blending convexity.
bl_cvxty_ents	Text specifies blending convexity.
bl_ed_convexity	Text specifies blending convexity of an entity.
bl_how_ents	Text specifies what type of blending is to be performed at ends.

A

color	Number specifies the color used. (Specified as integer in code).
cross_section_forms	Number specifies the type of cross section.
face_body_rel	Text specifies the relationship between face and body entities.
face_body_rel_ents	Text specifies the relationship between face and body entities.
merge_action	Text specifies how merging is performed.
rad_form_ents	Text specifies the type of radius blending.
sec_form_ents	Text specifies the form of section.
split_action	Text specifies what to do with elements involved in a split.
token_name	Capitalized text specifies token names. (static char* in code).
trans_action	Text specifies what to do with elements involved in a transform.
transition_ents	Text specifies what to do with entities involved in a transition.
underlying_sf_type	Text specifies surface type.

Save File Items

"1d"	Implements the ELEM1D.
"2d"	Implements the ELEM2D.
"A#"–"Z#"	Composes a law function that uses the identity law, x_n , to take and return the n -th input argument.
"ABS"	Composes a law mathematic function that takes the absolute value of another law mathematic function.
"AND"	Used with PIECEWISE to create a logical AND conditional.

"ARCCOS"	Composes a law mathematic function that finds the arc cosine.
"ARCCOSH"	Composes a law mathematic function that finds the inverse hyperbolic cosine.
"ARCCOT"	Composes a law mathematic function that finds the arc cotangent.
"ARCCOTH"	Composes a law mathematic function that finds the inverse hyperbolic cotangent.
"ARCCSC"	Composes a law mathematic function that finds the arc cosecant.
"ARCCSCH"	Composes a law mathematic function that finds the inverse hyperbolic cosecant.
"ARCSEC"	Composes a law mathematic function that finds the arc secant.
"ARCSECH"	Composes a law mathematic function that finds the inverse hyperbolic secant.
"ARCSIN"	Composes a law mathematic function that finds the arc sine.
"ARCSINH"	Composes a law mathematic function that finds the inverse hyperbolic sine.
"ARCTAN"	Composes a law mathematic function that finds the arc tangent.
"ARCTANH"	Composes a law mathematic function that finds the inverse hyperbolic tangent.
"BEND"	Creates a law to bend from a position around an axis in a given direction a specified amount.
"COS"	Composes a law mathematic function that finds the cosine.
"COSH"	Composes a law mathematic function that finds the hyperbolic cosine.
"COT"	Composes a law mathematic function that finds the cotangent.

"COTH"	Composes a law mathematic function that finds the hyperbolic cotangent.
"CROSS"	Composes a law mathematic function that takes the cross product of two laws.
"CSC"	Composes a law mathematic function that finds the cosecant.
"CSCH"	Composes a law mathematic function that finds the hyperbolic cosecant.
"CUR"	Composes a law mathematic function that returns the positions of the defining curve.
"CURC"	Composes a law mathematic function that returns the curvature of the curve at a parameter value.
"D"	Composes a law mathematic function that takes one or more derivatives of a given law with respect to a given variable.
"DCUR"	Composes a law mathematic function that finds the derivative of a given curve.
"DOT"	Composes a law mathematic function that takes the dot product of two vectors.
"DSURF"	Composes a law mathematic function that finds the derivative of a given surface.
"DWIRE"	Composes a law mathematic function that finds the derivative of a given wire.
"E"	Provides the representation for ϵ to the accuracy of the system.
"EDGE#"	Composes a law function with a tag for an edge or bounded curve used as an input argument.
"EXP"	Composes a law mathematic function that takes e to a given power.
"FRENET"	Composes a law mathematic function that returns the second geometric derivative of its sublaw.

“LN”	Composes a law mathematic function that takes the log base e (or the natural log) of the given value.
“LOG”	Composes a law mathematic function that takes the log of a given base of the given value.
“MAX”	Composes a law mathematic function that finds the maximum of two or more input laws.
“MIN”	Composes a law mathematic function that finds the minimum of two or more input laws.
“MINROT”	Composes a law mathematic function that returns the minimum rotation.
“NORM”	Composes a law mathematic function that normalizes a law.
“NOT”	Used with PIECEWISE to create a logical NOT conditional.
“O”	Creates function composition, as in “f of g”, where f and g are both law mathematic functions.
“OR”	Used with PIECEWISE to create a logical OR conditional.
“PI”	Provides the representation for pi to the accuracy of the system.
“PIECEWISE”	Permits laws to evaluate differently based on conditional definition statements.
“ROTATE”	Composes a law mathematic function that transforms vectors.
“SEC”	Composes a law mathematic function that finds the secant.
“SECH”	Composes a law mathematic function that finds the hyperbolic secant.
“SET”	Composes a law mathematic function that returns a 1 if its sublaw is positive and 0 if its sublaw is negative or zero (0).

"SIN"	Composes a law mathematic function that finds the sine.
"SINH"	Composes a law mathematic function that finds the hyperbolic sine.
"SIZE"	Returns the square root of the sum of the squares of a given vector (e.g., VEC) elements.
"SQRT"	Composes a law mathematic function that takes the square root of a given law.
"STEP"	Composes a law mathematic function that defines functions with disjoint intervals.
"SURF"	Composes a law mathematic function that returns the positions of the defining surface.
"SURF#"	Composes a law function with a tag for a surface used as an input argument.
"SURFNORM"	Composes a law mathematic function that returns the normal to a surface at a given position.
"SURFPERP"	Composes a law mathematic function that returns the position on a surface of point projected perpendicular to surface.
"SURFVEC"	Composes a law mathematic function that returns a parameter vector on a surface.
"T"	Composes a law function that uses the identity law to take and return the first input argument.
"TAN"	Composes a law mathematic function that finds the tangent.
"TANH"	Composes a law mathematic function that finds the hyperbolic tangent.
"TERM"	Composes a law mathematic function that returns a single term from a given multi-dimensional function.
"TRANS"	Composes a law mathematic function that transforms positions.

“TRANS#”	Composes a law function with a tag for a transform used as an input argument.
“TWIST”	Composes a law mathematic function that returns a twisted vector field about a given path.
“U”	Composes a law function that uses the identity law to take and return the first input argument.
“UNBEND”	Creates a law to unbend from a position around an axis in a given direction a specified amount.
“V”	Composes a law function that uses the identity law to take and return the second input argument.
“VEC”	Composes a law mathematic function that is a vector of arbitrary dimensions.
“WIRE”	Composes a law mathematic function that returns the positions of the defining a wire.
“WIRE#”	Composes a law function with a tag for a wire used as an input argument.
“X”	Composes a law function that uses the identity law to take and return the first input argument.
“Y”	Composes a law function that uses the identity law to take and return the second input argument.
“Z”	Composes a law function that uses the identity law to take and return the third input argument.
“adjedge”	Records blank body edges that are in contact with the sheet boundary, which are called <i>adjacent edges</i> .
“adjface”	Records blank body faces that are in contact with the sheet boundary, which are called <i>adjacent faces</i> .
“adv_var_blend”	Defines the blend attribute for edge sequence-following blends.
“agc”	Identifies that the attribute came from the ATTRIB_AGC class.

"attrib"	Represents common data and functionality for all attributes.
"attrib_fhl"	Faceted Hidden Line Component organizational class.
"attrib_fhl_slist"	Attaches FHL line segment data to model entities.
"attrib_fhlgeom"	Attaches FHL geometry data to model entities as attributes.
"attrib_fhlhead"	Heads the list of FHL attributes attaching hidden line data to the model.
"attrib_fhlmark"	Detects changes to underlying geometry.
"attrib_fhlplist"	Attaches FHL polyline data to model ENTITYs.
"attrib_var_blend"	Defines the blend attribute for variable radius blends.
"bk"	Links surface elements to curve elements.
"bl_cr"	Implements several trivial constant round geometries for edge-face and some face-face cases.
"bl_ent"	A base class for a blend on some unspecified entities.
"bl_ent_ent"	A base class for entity-entity blends.
"bl_inst"	Defines the instruction attributes that hang onto the edges and vertices of a blend sequence, that instruct the blend algorithm.
"bl_seg"	Defines the instruction attributes that hang onto the coedges and vertices of a sheet face to make the correct efinit and faceint attributes.
"bl_support"	Derived class for the blend support point.
"bl_taned"	Defines a variable radius entity-entity blend.
"bl_vr"	Defines a variable radius entity-entity blend.
"bldcur"	This is one long edge of a general blend surface – a spring curve.

“blend”	Defines the basic blend attribute from which derive specific blend attributes.
“blendsupcos”	Derived class for the curve–on–surface case.
“blendsupcur”	Derived class for the blend support curve.
“blendsuppnt”	Derived class for the blend support point.
“blendsupsur”	Derived class for the blend support surface.
“blendsupzro”	Derived class for the blend support zero curve.
“blinfo”	Defines attributes used internally by the blending algorithm to record intermediate results.
“blndsprngcur”	This is one long edge of a general blend surface – a spring curve.
“body”	Represents a wire, sheet, or solid body.
“box”	For internal use only.
“bulletin”	Holds the changes to a single ACIS entity.
“bulletin_board”	Holds all bulletins resulting from an API call.
“camera”	Used to save camera data in an ACIS part file.
“cap_ext”	For internal use only.
“cap_info”	For internal use only.
“capping_record”	For internal use only.
“cell”	Attaches the subportion of a lump.
“cell2d”	Connects the faces of a sheet.
“cell3d”	Connects subportion of a lump.
“cell_ptr”	Implements an attribute describing a cell in the Cellular Topology Component.
“cface”	A reference to one <i>side</i> of a FACE.
“cface_col_att”	Assigns a color to a volume.
“cface_ptr”	Defines an attribute that records two CFACE entities that refer to a given face.

"cface_vol_att"	Used for volume attribute propagation member functions.
"chamfer"	Defines a flat chamfer blend.
"coedge"	Relates EDGES with adjacent EDGES and owning ENTITYs.
"coinvert"	Creates a coincident vertex attribute for stitching.
"colour"	User attribute class definition for a color.
"comment"	Constructs a PTC attribute.
"compcurv"	Records a composite curve as a (lowercase) compcurv.
"cone"	Identifier used by more than one class.
"const_chamfer"	Defines a flat chamfer blend.
"const_round"	Defines a circular rolling-ball blend of constant radius.
"convexity"	For internal use only.
"copy_marker"	For internal use only.
"crv_cstrn"	Curve constraints.
"crv_load"	Curve loads.
"crvcrvblndsurr"	Implements the variable-radius edge-edge blend surface.
"crvsrflndsurr"	Implements the variable-radius edge-face blend surface.
"cs"	Implements the mesh curve and surface link attributes.
"cshell"	One connected portion of a cell's boundary.
"ct"	Defines a base attribute for the Cellular Topology Component.
"curve"	Identifier used by more than one class.

"d5c2_cur"	Defines an interpolated curve subtype which is a subtype of an int_cur.
"dc_2acis"	Connects the deformable surface functions to the ACIS modeler.
"degenerate_curve"	Used for skinning and lofting surfaces.
"delete"	For internal use only.
"delta_state"	Holds all bulletin boards resulting from modification to a model.
"dict_entry"	Stores names and their corresponding entities in an unordered linked list.
"display_attribute"	Defines DISPLAY_ATTRIB attribute to link ENTITYs with their display.
"dispose"	Makes a face disposal attribute.
"dist_press"	Distributed pressure.
"dm_2acis"	Connects the deformable surface functions to the ACIS modeler.
"ds_2acis"	Connects the deformable surface functions to the ACIS modeler.
"ds_cstrn"	Deformable surface constraint.
"ds_group"	Connects the deformable surface functions to the ACIS modeler.
"ds_load"	Deformable surface load.
"ds_model"	Deformable surface model.
"ecoin"	Defines the attribute to be attached to edges of the sheet, where they are coincident with edges of the blended body.
"edge"	Represents a physical edge.
"edgetapersur"	Evaluator for a general edge tapered surface.
"efint"	Attached to bodies during the first stage of Booleans.

“elem”	Implements the ELEM class.
“ellipse”	Identifier used by more than one class.
“elliptical”	Represents data for an elliptical variable radius blend.
“entity_attrib”	Defines a generic attribute that owns an entity.
“entity_ptr”	Describes a pointer entity as a derived class of ENTITY so that these entities are logged and rolled back and forth together with models.
“error_entity”	For internal use only.
“exactcur”	Represents an exact spline curve.
“exactsur”	Represents an exact spline surface.
“expblend”	Implements a derived blend attribute for marking blend sheet faces to indicate their provenance.
“exppc”	Defines an explicit parameter-space curve.
“eye”	A base class for classes used in the Faceter Component .
“eye_refinement”	Controls the accuracy and types of polygons generated in the Faceter Component .
“face”	Represents a bounded portion of a SURFACE.
“faceint”	Defines an attribute to record the intersection of a face of one body with a face of the other body during a Boolean operation.
“fctd_mark”	Marks faceted faces.
“ffblend”	Defines a general face-face blend.
“fixed_width”	Represents data for a variable radius blend with a fixed width.
“fmesh”	Identifies that a mesh is attached to the entity.

“functional”	Represents data for a variable radius blend using a radius defined by a function.
“gen”	Defines the organization attribute class for the
Generic Attributes	
Component.	
“group”	Entity class that points to a list of entities.
“group_ptr”	Implements an attribute that points to a list of entities.
“history”	Defines an attribute for the history stream mechanism.
“history_stream”	Contains pointers to the initial delta state, to the last noted active delta state, and to the delta state under construction.
“id_attribute”	Defines class ID_ATTRIB to allow identification of ENTITYs in a table.
“imppc”	Defines an implicit parameter-space curve.
“int”	Used to translate mesh surfaces to and from external file formats.
“intcoed”	Defines an attribute for linking edges and faces.
“intcurve”	Identifier used by more than one class.
“intedge”	Defines an attribute for linking intersection edges with the intersecting entities.
“integer_attrib”	Defines a generic attribute that contains an integer value.
“intercept”	For internal use only.
“intgraph”	Defines an attribute for classifying shells and lumps of two bodies participating in a Boolean operation.
“invert”	Defines an attribute for linking graph vertices with the intersection record(s) giving rise to them.

"lawintcur"	Defines a curve from a law.
"lawsur"	Creates a surface using a law.
"link"	Used to translate mesh surfaces to and from external file formats.
"loop"	Bounds a FACE.
"lop_coedge"	Creates a coedge specific to local operations.
"lop_curve_ext_att"	Creates an attribute specific to local operations.
"lop_edge"	Creates an edge specific to local operations.
"lop_edge_att"	Creates a local operations attribute for an edge.
"lop_edge_cvxty_att"	For internal use only.
"lop_loop_attr"	Creates a coedge specific to local operations.
"lop_loop_class_att"	For internal use only.
"lop_protected_list_att"	For internal use only.
"lop_surface_ext_att"	Creates an attribute specific to local operations.
"lop_vert_att"	Creates an attribute specific to local operations.
"lop_vertex"	Creates a vertex specific to local operations.
"lopt_copy_att"	For internal use only.
"lump"	Represents a bounded, connected portion of space.
"meshsurf"	Implements the MESHSURF subclass of the SURFACE class.
"ms"	Represents attribute base class for the Mesh Surface Component .
"name_attrib"	Defines a named attribute for the Generic Attributes Component .
"named_attribute"	Defines class NAMED_ATTRIB to allow generic named attributes.

"named_int_attribute"	Defines class NAMED_INT_ATTRIB to provide named attributes with integer values.
"named_logical_attribute"	Defines class NAMED_LOGICAL_ATTRIB to provide named attributes with logical values.
"named_pos_attribute"	Defines class NAMED_POS_ATTRIB to provide named attributes with position values.
"named_real_attribute"	Defines class NAMED_REAL_ATTRIB to provide named attributes with real values.
"named_string_attribute"	Defines class NAMED_STRING_ATTRIB to provide named attributes with string values.
"named_vec_attribute"	Implements class NAMED_VEC_ATTRIB to allow named vector attributes.
"netsur"	Defines a net surface between a list of curves.
"node"	Identifier used by more than one class.
"norender_attribute"	Attribute to mark items not to be rendered.
"offintcur"	Represents a spline curve obtained by the intersection of two surfaces that are offsets of the given surfaces.
"offrel"	Implementation of the offset relation attribute.
"offsetintcur"	Represents an offset curve, which is offset in any plane.
"offsetvbsur"	Represents an offset vertex blend surface.
"offsur"	Represents the offset of a surface.
"offsetintcur"	Represents the offset of a curve lying on a surface.
"orthosur"	Creates an orthogonal surface.
"p1"	Implements one dimensional parameter nodes.
"p2"	Implements parameter nodes.
"parasil"	Creates an interpolated curve subtype which can precisely represent a parallel-view silhouette curve.

“parcur”	Represents a spline curve as a parameter curve on a spline surface for a parameterization.
“pcurcur”	Defines an interpolated curve subtype that is the 3D extension of the parameter curve representing a curve on a surface.
“pcurve”	Identifier used by more than one class.
“perspsil”	Creates an interpolated curve subtype which can precisely represent a perspective-view silhouette curve.
“phl”	Defines the phl tag identifier.
“phl_camera”	Defines a camera viewpoint against which hidden lines are calculated.
“phl_edge”	Defines a regular or silhouette edge.
“phl_segment”	Defines a line segment with visibility information.
“phl_vw”	Attaches hidden line data and viewing parameters to BODYs.
“pid_name”	Persistent identifier data attribute.
“pipesur”	A surface that is the envelope of a fixed-radius circle.
“plane”	Identifier used by more than one class.
“point”	Represents the position of a VERTEX as an object in the model.
“pointer_attrib”	Defines a generic attribute that contains a reference to an entity.
“position_attrib”	Defines a generic attribute that contains a position.
“projcur”	Implements an interpolated curve subtype to represent the perpendicular projection of a curve to a surface.
“pt_cstrn”	Point constraint.

"pt_press"	Point pressure.
"ptlist"	Associates a cyclic doubly linked list with the entity.
"rbase"	Defines a rendering attribute that holds information about surface visual attributes, sidedness, and texture space.
"rbblnsur"	This class implements the constant radius rolling ball blend surface.
"real_attrib"	Defines a generic attribute that contains a real value.
"ref"	This references a previously defined subtype in the save file.
"ref_vt"	Attaches REFINEMENT and VERTEX_TEMPLATE instances to other entities.
"rem_edge"	For internal use only.
"rem_protected_list_att"	For internal use only.
"rem_vertex"	For internal use only.
"render"	Defines an attribute that contains rendering information.
"rgb_color"	Defines a ATTRIB_RGB color attribute.
"rh_background"	Defines a background.
"rh_entity"	Provides common methods and data for other rendering classes.
"rh_env_map"	Defines an environment map.
"rh_foreground"	Defines a foreground.
"rh_light"	Defines a light source.
"rh_material"	Defines a material consisting of color, displacement, reflectance, and transparency.
"rh_texture_space"	Defines a texture space.

"rotsur"	Represents a surface of rotation.
"round"	Defines a circular rolling-ball blend.
"ruledtapersur"	Class to describe a surface tapered about an edge by a constant angle relative to a draft angle.
"sfcvfreeblnds"	Implements the variable-radius surface-curve/free blend surface.
"sg"	Organization base attribute class for the SG Husk.
"shadowtapersur"	Class to describe a surface that is tapered about a silhouette by a constant angle determined by a draft direction.
"shell"	Bounds a LUMP peripherally, or as an internal void.
"skinsur"	Defines a skin surface between a list of curves.
"skinsur2"	Defines a skin surface between a list of curves.
"sphere"	Identifier used by more than one class.
"spline"	Identifier used by more than one class.
"split"	Attached to each edge of each body which has a graph vertex properly within it.
"spring"	Marks edges lying on spring curves so they may be specially handled later.
"spring_load"	For internal use only.
"spring_set_load"	Set of spring loads.
"srfsrflnds"	Implements the variable-radius face-face blend surface.
"st"	Organization attribute from which various color, display, id, and other attributes are derived.
"sti_elat_attr"	Creates a temporary attribute used in sweeping.
"sti_nor_attr"	Creates a temporary attribute used in sweeping.

"sti_prof_attr"	Creates a temporary attribute used in sweeping.
"sti_psplrit_attr"	Creates a temporary attribute used in sweeping.
"sti_rel_attr"	Creates a temporary attribute used in sweeping.
"sti_sect_attr"	Creates a temporary attribute used in sweeping.
"sti_vlat_attr"	Creates a temporary attribute used in sweeping.
"straight"	Identifier used by more than one class.
"string_attr"	Defines a generic attribute that contains a string value.
"stripc"	Identifier used by more than one class.
"subsetintcur"	Represents a subset of a longer curve.
"subshell"	Represents a subdivision of a SHELL or SUBSHELL.
"subsur"	Represents the geometry of a spline surface, which is a subset region of another spl_sur.
"sumsur"	Represents a linear sum of two curves.
"supercell"	Identifies a grouping of cells or inferior supercells.
"surface"	Identifier used by more than one class.
"surfcur"	Represents a spline curve projected onto a surface within the given fit tolerance.
"surfintcur"	Represents the spline curves obtained from the intersection of two surfaces.
"sweepsur"	Defines the perpendicular sweep of a planar profile curve along a path curve.
"swepttapersur"	Class to describe a swept-tapered surface, in which a surface is tapered about an edge by a constant angle relative to a draft angle.
"sys"	Defines a base class for the Kernel Component.
"t3"	Implements planar triangular elements.

"tan_xedge"	Attaches an attribute to the cross edges of the blend sheet.
"tapersil"	Creates an interpolated curve subtype which can precisely represent a silhouette curve formed by applying a taper.
"tapersur"	Creates an edge-tapered surface.
"text_ent"	Routine to restore a TEXT_ENT entity from file.
"torus"	Identifier used by more than one class.
"transform"	Represents an overall transformation applied to a BODY.
"tri3sur"	Represents a mesh surface consisting only of planar triangular elements.
"tsl"	Defines a base class for a specific application developer.
"tubesur"	A surface that is the envelope of a fixed-radius circle.
"two_ends"	Represents data for a blend using a variable radius at both ends.
"undefc"	Identifier used by more than one class.
"units"	Specifies the units a model is defined in.
"unknown"	Represents common data and functionality that is mandatory in all classes that are permanent objects in the model.
"varblendsplsur"	Implementation of the base class for variable radius and other nonpipe blends. Derived from blend_spl_sur.
"vblend"	Defines the vertex blend attribute.
"vector_attr"	Defines a generic attribute that contains a vector.
"vertdge"	Contains a list of edge pointers.

“vertex”	Represents an end of an EDGE.
“vertex_template”	Represents the data to be generated at a facet node.
“vertexblendsur”	Defines the vertex blend surface class.
“wcs”	Defines the WCS class.
“wire”	Represents a collection of EDGES.
“xedge”	Attaches to cross edges of the blend sheet, recording the blended edge giving rise to the face on one side of the cross edge.
“xvert”	Implements the derived blend attribute for flagging a blend sheet pointed vertex.
“xverted”	Defines an attribute to be attached to point vertices of the vertex blend sheet.
ATTRIB_NODE	Represents a grazing touch of an intersection curve with the side (boundary) of an element.
ATTRIB_PHL	Defines the phl tag identifier.
BDY_GEOM	The base class for different representations of the boundary.
BDY_GEOM_CIRCLE	Implements the boundary geometry as an ellipse (actually a circle).
BDY_GEOM_DEG	This class implements a boundary section as a degenerate point.
BDY_GEOM_PCURVE	Defines all the functions the underlying geometry of the boundary curves must provide.
BDY_GEOM_PLANE	Implements the boundary geometry as a curve on a surface which must be a plane.
CONE (1) class	Defines a cone as an object in the model.
CURVE (1) class	Represents a variety of curve geometries as an object in the model.
ELLIPSE (1) class	Defines an ellipse as an object in the model.
ENTITY	Represents common data and functionality that is mandatory in all classes that are permanent objects in the model.

ENTITY_PHL	Defines the phl tag identifier.
INTCURVE (1) class	Defines an intersection curve as an object in the model.
NODE	Represents a grazing touch of an intersection curve with the side (boundary) of an element.
PCURVE (1) class	Defines a 2D parameter-space approximation to a CURVE as an object in the model.
PLANE (1) class	Defines a plane as an object in the model.
SPHERE (1) class	Defines a sphere as an object in the model.
SPLINE (1) class	Defines a parametric surface as an object in the model.
STRAIGHT (1) class	Defines an infinite line as an object in the model.
STRIPC (1) class	Records a parametric surface as a STRIPC .
SURFACE (1) class	Defines a generic surface as an object in the model.
TORUS (1) class	Defines a torus as an object in the model.
UNDEFC (1) class	Defines a curve that is undefined except for its end points.
af_node_mapping	Lookup table interface for token types in mesh nodes.
blend_spl_sur	This class is used as a base class for all blend surfaces.
blend_support	Represents the geometry on which a spring curve of a blend lies.
bounded_curve	Defines a bounded curve.
bs2_curve_def	Defines a 2D B-spline curve.
bs3_curve_def	Provides an interface between ACIS and the underlying spline library.
bs3_surface_def	Provides an interface between ACIS and the underlying spline library.

com_cur	Represents of a patchwork curve used to represent boundaries of a meshsurf , which is a patchwork surface.
compcurv (2) class	Records a composite curve as a (lowercase) compcurv .
cone (2) class	Defines the elliptical single cone.
constant	Composes a law function that is a constant number.
curve (2) class	Derives specific data used for the curve classes.
curve type	Defines more specifically the type of curves.
discontinuity_info	Stores discontinuity information for a curve or surface.
division	Composes a law mathematic function that uses the division (“/”) operator.
ellipse (2) class	Defines an ellipse.
equal	Used with PIECEWISE to create a logical = conditional.
exponent	Composes a law mathematic function that uses the exponentiation, or power, (“^”) operator.
greater_than	Used with PIECEWISE to create a logical > conditional.
greater_than_or_equal	Used with PIECEWISE to create a logical >= conditional.
int_cur	Defines spline curves.
intcurve (2) class	Represents parametric object-space curves that map an interval of the real line into a 3D real vector space (object-space).
law	Creates the base class for the derived law classes.
law_data	Creates a wrapper for other ACIS classes for passing as arguments to laws.

less_than	Used with PIECEWISE to create a logical < conditional.
less_than_or_equal	Used with PIECEWISE to create a logical <= conditional.
minus	Composes a law mathematic function that uses the minus, or subtraction (“-”) operator.
msh_sur	Records a composite mesh surface.
negate	Composes a law mathematic function that uses the unary minus, or negation (“-”) operator.
not_equal	Used with PIECEWISE to create a logical != conditional.
path#	Composes a law function with a tag for a path (e.g., edge or wire) used as an input argument.
pcurve (2) class	Defines a 2D curve defined in the parameter space of a parametric surface.
pcurve type	Defines a 2D curve defined in the parameter space of a parametric surface.
plane (2) class	Defines a planar surface.
plus	Composes a law mathematic function that uses the addition (“+”) operator.
sphere (2) class	Defines a spherical surface.
spl_sur	Defines an abstract base class from which spline surface definitions are derived.
spline (2) class	Records a B-spline surface.
straight (2) class	Defines an infinite straight line represented by a point and a unit vector specifying the direction.
stripc (2) class	The strip curve (stripc) is a surface defined in a neighborhood of and passing through a given object-space curve.
subtype_object	Defines the master object from which all subtype objects must be derived.

surface (2) class	Base class for all ACIS surface types that defines the basic virtual functions that are supplied for all specific surface classes.
surface type	More detailed definition of surface.
times	Composes a law mathematic function that uses the times or multiplication (“*”) operator.
torus (2) class	Represents tori.
undefc (2) class	Denotes a curve that is undefined except for its end points, for which there are explicit positions, directions, and curvatures.
var_cross_section	Represents the cross section of a blend surface.
var_radius	Defines variable radius information for a variable radius blend.