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

ACIS_OBJECT	Provides a base class for class level memory management of ACIS classes.
base_configuration	Initializes the Base library.
complex_number	Creates a data structure for the manipulation of complex number.
enum_table	Defines objects for storing the mapping between the enum values and their string representation.
error_info	Defines objects for returning ACIS error information.
error_list_info	Chains a list of error_infos together.
exit_callback	Executes standard exit for ACIS.
input_callback	Creates the standard input for ACIS.
message_module	Contains all messages for a module.
option_header	Records a value that denotes whether the option is *on*, *off*, or set to a given value.
output_callback	Creates output callback standard output for ACIS.

SPAbox . . . . . Represents a bounding box.

SPAinterval . . . . . Records an interval on a line.

SPAmatrix . . . . . Defines a 3x3 affine transformation acting on vectors and positions.

SPAnvector . . . . . Implements an *n* dimensional vector.

SPAparameter . . . . . Defines a curve parameter value.

SPApar_box . . . . . Defines a bounding box in parameter space by four values of class parameter: low_u, high_u, low_v, high_v.

SPApar_dir . . . . . Defines a direction vector (du,dv) in 2D parameter–space.

SPApar_pos . . . . . Defines a parameter position in the parameter-space of a surface.

SPApar_transf . . . . . Defines a parameter space transformation containing scaling and translation components.

SPApar_vec . . . . . Defines a vector (*du*, *dv*) in 2D parameter-space.

SPAposition . . . . . Represents position vectors (points) in 3D Cartesian space that are subject to certain vector and transformation operations.

SPAtransf . . . . . Represents a general 3D affine transformation.

SPAunit_vector . . . . . Provides a direction in 3D Cartesian space that has unit length.

SPAvector . . . . . Represents a displacement vector in 3D Cartesian space.

toolkit_callback . . . . . Defines the toolkit_callback base calls and the toolkit_callback_list class.

toolkit_callback_list . . . . . Stores the list of toolkit_callbacks.

VOID_LIST . . . . . Creates a variable-length list of void*'s.

Enumerations

Topic: Ignore

AcisMemType . . . . . Specifies the persistence of memory allocated via the ACIS memory manager.

Functions

Topic: Ignore

antiparallel . . . . . Determines if two vectors are anti–parallel (within some resolution).

are_parallel . . . . . Determines if two vectors or two unit vectors are parallel within SPAresnor.

are_perpendicular . . . . . Determines if two vectors or two unit vectors are perpendicular (within SPAresnor).

biparallel . . . . . Determines if two vectors are bi–parallel (within some resolution).

collapse_all_free_lists . . . . . Returns all unused blocks of memory to the operating system.

coordinate_transf . . . . . Constructs a coordinate transformation.

debug_real . . . . . Prints to the output file a real sometimes specified with the given title.

distance_to_line . . . . . Determines the distance from a position to a line.

distance_to_plane . . . . . Determines the distance from a position to a plane.

distance_to_point . . . . . Determines the distance between two points.

find_err_ident . . . . . Gets the identifier string for an error number. Returns “UNKNOWN” if the error number is invalid.

find_err_mess . . . . . Gets the message string for an error number. Returns “unknown error” if the error number is invalid.

find_option . . . . . Gets the given option in the list.

get_major_version . . . . . Returns the major version number of the ACIS executable.

get_minor_version . . . . . Returns the minor version number of the ACIS executable.

get_option_list . . . . . Returns pointer to the head of the option list.

get_resabs . . . . . Gets the SPAresabs resolution.

get_resfit . . . . . Gets the SPAresfit resolution.

get_resmch . . . . . Gets the resmch resolution.

get_resnor . . . . . Gets the SPAresnor resolution.

initialize_base . . . . . Initializes the Base library.

interpolate . . . . . Interpolates between two positions.

is_equal . . . . . Determines if two values are equal.

is_greater_than . . . . . Determines if the first value is greater than the second value.

is_less_than . . . . . Determines if the first value is less than the second value.

is_negative . . . . . Determines if a value is negative.

is_on_line . . . . . Determines if a SPAposition is on a line within SPAresabs.

is_on_plane . . . . . Determines if a SPAposition is on a plane within SPAresabs.

is_positive . . . . . Determines if a value is positive.

is_zero . . . . . Determines if a value is zero relative to SPAresabs.

normalise . . . . . Converts a vector into a unit vector.

parallel . . . . . Determines if two vectors are parallel (within some resolution).

perpendicular . . . . . Determines if two vectors are perpendicular (within some resolution).

reflect_transf . . . . . Constructs a reflection transformation for a reflection about the plane through the perpendicular to the given vector.

rotate_transf . . . . . Constructs a rotation transformation for a simple rotation by an angle about the given vector.

same_matrix . . . . . Determines if two matrices are equal, given some tolerance.

same_vector . . . . . Determines whether or not two vectors are the same.

scale_transf . . . . . Constructs a transformation for overall scaling.

scaling . . . . . Creates the matrix for scaling.

terminate_base . . . . . Terminates the Base library.

translate_transf . . . . . Constructs a translation transformation to translate along the given vector.

Options

Topic: Ignore

crash . . . . . Sets crash on error.

debug_absolute_addresses . . . . . Controls how memory address for pointers are returned for debugging: as absolute or relative, and as hex or long integer.

mmgrfile . . . . . Sets the name of the file to which to write memory manager debugging and memory leak auditing information.

mmgrfill . . . . . Enables or disables pattern filling of unallocated memory by the memory manager.

mmgrlog . . . . . Enables or disables writing of memory manager debugging and memory leak auditing information to a file.

stack_check_error . . . . . Controls behavior when the stack limit is reached.

timing . . . . . Sets whether or not timing information is returned.

Typedefs

Topic: Ignore

err_mess_type . . . . . Identifier for an error string.