Chapter 7. Debugging

Topic: *Debugging

The following general ACIS application debugging techniques exist:

- ENTITY class debug method
- Error and warning messages
- Run-time debugging
- Scheme debugging extensions

ENTITY Class Debug Method

Topic: *Debugging

The ENTITY class (and each class derived from it) contains a method called debug_ent that is called to dump the instance's data for debugging. The method ENTITY::debug_ent is implemented as:

```
void ENTITY::debug_ent(FILE* fp) const
{
        // Start with this entity's identifier

        debug_header(this, fp);

        // Now the entity data.

        if (fp != NULL) {
            debug_title("Rollback pointer", fp);
            debug_pointer(rollback_ptr, fp);
            debug_newline(fp);
        }

        debug_new_pointer("Attribute list", attrib(), fp);

        // Put out anything from the unknown text.

        text_ptr->debug_ent(fp);
}
```

This method calls several functions that dump data using fprintf. For example:

```
void debug_title(char const* title, FILE* fp)
{
        if (fp != NULL) {
            if (title == NULL)
                title = ””;
            fprintf(fp, ”\t%–16.16s: ”, title);
        }
}
```

The following general debug routines are used by the debug_ent methods to print out various types of values:

debug_dist	Prints a real representing a signed distance. It is considered to be zero if its magnitude is less than SPAresabs.
debug_newline	Prints a new line character.
debug_norm	Prints a real representing a normalized, dimensionless quantity. It considered to be zero if its magnitude is less than SPAresnor.
debug_pointer	Prints a pointer. By default, prints this as a relative address. If the option debug_absolute_addresses is on, an absolute address is used.
debug_pointer_str	Prints a pointer as a string. By default, prints this as a relative address. If the option debug_absolute_addresses is on, an absolute address is used.
debug_real	Prints a real number with appropriate precision.
debug_time	Prints the debugging time difference. This is the amount of time since the last call to this function or to the debug_time_init function.
debug_time_init	Initializes the debugging time.

Error and Warning Messages

Topic: *Debugging

The following C++ functions can be called by applications to find and print error messages once an error has occurred (these and other error handling functions are described in Chapter 6, *Error Handling and Messaging*):

find_err_entry	find_err_ident	find_err_mess
find_err_module	print_warnerr_mess	get_warnings
init_warning		

Run–Time Debugging

Topic: *Debugging

Many software environments provide interactive run-time debuggers that allow you to stop execution of the program, examine variables, call functions, etc. (for example, in the UNIX environment a common debugger is called *dbx*).

Several functions have been defined in the ACIS software to aid debugging with such a debugger. These functions are not linked into the system by default, because they are not called by any ACIS functions. To access these functions, you must force the linkage of the file kern/kernel/sg_husk/debug/sg_debug.cxx. The functions provided in this file include:

dbuvec	Debug a SPAunit_vector
dbvec	Debug a SPAvector
dbpos	Debug a SPAposition
dbtransf	Debug a SPAtransf
dbcurve	Debug a curve
dbpcurve	Debug a pcurve
dbsurface	Debug a surface
dbbs2_curve	Debug a bs2_curve
dbbs3_curve	Debug a bs3_curve
dbbs3_surface	Debug a bs3_surface
dbedge	Print info about an EDGE
dbcoedge	Print info about a COEDGE
dbbs2c	Print info about a bs2_curve
dbbs3c	Print info about a bs3_curve
bs2_curve_step_eval	Evaluate a bs2_curve at increments

bs3_curve_step_eval	Evaluate a bs3_curve at increments
dblaw	Debug a law
dbent	Debug an ENTITY
dbentall	Debug this ENTITY and its siblings and children
dbentallsz	Get the size for this ENTITY and its siblings and children
dbentkids	Debug this ENTITY and its children
dbentkidsz	Get the size for this ENTITY and its children
dbsave	Save an ENTITY to dbfile
dblist	Debug an ENTITY_LIST
dblistent	Debug an ENTITY_LIST
dbdbg	Turn on debugging module
dbopti	Pass integer to option
dboptd	Pass double to option
dboptc	Pass string to option

The following output streams are used by these functions:

dbfile	Debug output stream stream
dbout	stdout output stream
dberr	stderr output stream

Scheme Debugging Extensions

Topic: *Debugging

Several Scheme extensions are provided to facilitate debugging. These include:

debug:all–modules	Sets the debug level for all modules.
debug:file	Sets the debug output device.
debug:list–modules	Displays the current debug level for all modules.
debug:module	Sets the debugging level for a particular module.