

Chapter 1.

Introduction to Fundamental Concepts

Topic:

Ignore

ACIS 3D Geometric Modeler (ACIS) is an object-oriented three-dimensional (3D) geometric modeling engine from *Spatial Corp.* (*Spatial*). It is designed for use as the geometry foundation within virtually any end user 3D modeling application. ACIS provides an open architecture framework for wireframe, surface, and solid modeling from a common, unified data structure.

ACIS is designed using software component technology. ACIS is written in C++ and consists of a set of C++ classes (including data and member functions, or methods) and functions. A developer uses these classes and functions to create an end user 3D application.

Spatial does not guarantee that an operation on an object in ACIS will have the same result from release to release.

In This Manual

This manual discusses some fundamental concepts of ACIS, including:

- How ACIS uses C++
- Basic math concepts such as dimensionality, continuity, transforms, etc., and how ACIS implements a math foundation in C++
- Model objects (including the C++ class ENTITY and its relationship to other classes)
- Construction and model geometry of ACIS models
- Model topology and boundary representation
- Curves and surfaces
- Tolerance variables in ACIS
- Model modification and analysis techniques

Prerequisites for Using ACIS

Before you begin working with ACIS, you should have the following foundation:

- An understanding of geometric modeling

- An understanding of object oriented design and programming
- Familiarity with C++ (preferably, prior programming experience with C++)

The following list summarizes the minimum set of ACIS fundamental subjects you should understand before you start to develop your application:

- ACIS software architecture, including the object libraries
- Geometry
 - mathematical C++ classes (such as SPAPosition, SPAvector, SPAtansf)
 - construction geometry C++ classes (such as curve, sphere, surface, etc.)
 - model geometry C++ classes (such as CURVE, SPHERE SURFACE, etc.)
- Model topology
 - boundary representation
 - topology C++ classes (such as BODY, EDGE, FACE, etc.)
- Entities and the ENTITY class
- Model objects and class relationships
- Attributes
 - types of attributes
 - C++ class ATTRIB and classes derived from it
- History and roll
 - bulletin board
 - delta state
 - history stream
- Save and restore
 - save file format
 - C++ class ENTITY_LIST

These subjects are discussed in the following manuals in online help (as well as in various reference templates for C++ functions and classes):

- *3D ACIS Getting Started Guide*
- *3D ACIS Fundamental Concepts Guide*
- *3D ACIS Application Development Manual*
- Component manuals, including the *Kernel Component Manual*