Chapter 3.
Functions

Topic: Ignore

The function interface is a set of Application Procedural Interface (API) and Direct Interface (DI) functions that an application can invoke to interact with ACIS. API functions, which combine modeler functionality with application support features such as argument error checking and roll back, are the main interface between applications and ACIS. The DI functions provide access to modeler functionality, but do not provide the additional application support features, and, unlike APIs, are not guaranteed to remain consistent from release to release. Refer to the *3D ACIS Online Help User's Guide* for a description of the fields in the reference template.

api_hollow_body

Function: Shelling

Action: Creates a thin walled solid shell from a solid body.

Prototype:

```
outcome api_hollow_body(
    BODY* body,              // body to be shelled
    int const nopenface,     // number of faces being
                             // omitted
    FACE* openface[],        // faces being omitted
    double thickness,        // distance to offset
    SPAposition box_low,     // start of intersection
                             // box to be used
    SPAposition box_high,    // end of intersection
                             // box to be used
    AcisOptions* ao          // ACIS options
        = NULL
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/top/body.hxx"
#include "kernel/kerndata/top/face.hxx"
#include "shl_husk/api/shl_api.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Shells the supplied body by the thickness distance omitting the specified faces which thus generate openings into the internal void.

Body faces with radial surfaces which cannot be offset by the thickness are removed and the resulting wound healed by the surrounding face surfaces. Radial surfaces are spheres, cones, and tori, blended edges and blended vertices.

This generally leads to shells which are locally overly thick. However in the case of vertex blends with large setbacks being removed, the shell may be locally too thin, or may even have a hole.

The box low and high arguments are used to limit any geometrical intersections and so improve performance. Limiting the intersection improves performance. The box must contain the final faces and if omitted defaults to the size box.

Multi–lump bodies may be hollowed, as long as at least one lump has only one shell. Lumps with more than one shell are not hollowed.

Mergeable edges will be retained if they have a NO_MERGE_ATTRIB.

Topology Changes:

Refer to the topology changes listed for the function, api_tweak_faces.

Geometry Changes:

Refer to the geometry changes listed for the function, api_tweak_faces.

Errors: In addition to the following, refer to the Errors listed for the function, api_tweak_faces.

Valid offset (greater than minus half the body box max side),
and not a zero offset (magnitude greater than twice SPAresabs) or error;–
LOP_OFF_BAD_OFFSET
Valid body transformation (no shear component) or error;–
LOP_BAD_BODY_TRANSFORM
Body must have at least one lump with only one shell, or error;–
LOP_HOL_MULTI_SHELL
Some faces must remain unopen, or error;–
LOP_HOL_ALL_OPEN
The model is too large or is positioned outside the modeling space
LOP_OFF_BAD_MODEL_EXTENTS

Limitations: Refer to the Limitations listed for the function, api_tweak_faces.

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: Changes model

api_hollow_body_specific

Function: Shelling

Action: Creates a thin walled solid shell from a solid body, with default and specific inner and outer offsets.

Prototype:

```
outcome api_hollow_body_specific (
    BODY* body,                  // body to be shelled
    int const nopenface,         // number of faces
                                 // being omitted
    FACE* openface[],            // faces being
                                 // omitted
    double inner_default_offset,// default inner
                                 // offset (usually –)
    double outer_default_offset,// default outer
                                 // offset (usually +)
    int const n_spec_inner,      // faces with
                                 // specific inner
                                 // offsets
    FACE* spec_inner_face[],     // specific inner
                                 // offsets
    double spec_inner_off[],     // specific inner
                                 // offsets
    int const n_spec_outer,      // no of specific
                                 // outer offsets
    FACE* spec_outer_face[],     // faces with
                                 // specific outer
                                 // offsets
    double spec_outer_off[],     // specific outer
                                 // offsets
    SPAposition box_low,         // start of
                                 // intersection box
                                 // to be used
    SPAposition box_high,        // end of
                                 // intersection box
                                 // to be used
    AcisOptions* ao = NULL       // ACIS options
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/top/body.hxx"
#include "kernel/kerndata/top/face.hxx"
#include "shl_husk/api/shl_api.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description:

Shells the supplied body by the offset distances omitting the specified open faces which thus generate openings into the internal void. The default offsets are applied to all faces except the open faces, unless overridden by specific offsets.

Body faces with radial surfaces which cannot be offset by the thickness are removed and the resulting wound healed by the surrounding face surfaces. Radial surfaces are spheres, cones, and tori, blended edges and blended vertices.

This generally leads to shells which are locally overly thick. However in the case of vertex blends with large setbacks being removed, the shell may be locally too thin, or may even have a hole.

The box low and high arguments are used to limit any geometrical intersections and so improve performance. Limiting the intersection improves performance. The box must contain the final faces and if omitted defaults to the size box.

Multi–lump bodies may be hollowed, as long as at least one lump has only one shell. Lumps with more than one shell are not hollowed.

Mergeable edges will be retained if they have a NO_MERGE_ATTRIB attached.

Topology Changes:

Refer to the topology changes listed for the function, api_tweak_faces.

Geometry Changes:

Refer to the geometry changes listed for the function, api_tweak_faces.

Errors: In addition to the following, refer to the Errors listed for the function, api_tweak_faces.

The inner or outer default must be set (size greater than twice SPAresabs), and valid (greater than minus half the body box max side), and the inner default offset must be less than the outer offset.

LOP_OFF_BAD_OFFSET

Specific offsets must be non zero (size greater than twice SPAresabs), and valid (greater than minus half the body box max side), and the outer offset for any face (default or specific) must be greater than the inner offset (default or specific).

LOP_BAD_BODY_TRANSFORM
Valid body transformation (no shear component).

LOP_HOL_MULTI_SHELL
Body must have at least one lump with only one shell.

LOP_HOL_ALL_OPEN
Some faces must remain unopen.

LOP_TWK_BAD_FACE
Open faces must not have a specific offset.

Limitations: Refer to the Limitations listed for the function, api_tweak_faces.

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: Changes model

api_initialize_shelling

Function: Shelling, Modeler Control

Action: Initializes the shelling library.

Prototype:
```
outcome api_initialize_shelling ();
```

Includes:
```
#include ”kernel/acis.hxx”
#include ”kernel/kernapi/api/api.hxx”
#include ”shl_husk/api/shl_api.hxx”
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: System routine

api_offset_faces_make_sheet

Function: Shelling

Action: Offsets an array of faces, (each face as required) and makes a new sheet out of the offset faces. The original body is untouched.

Prototype:

```
outcome api_offset_faces_make_sheet (
    int const& n_def_face,   // number of faces
                             // offset
    FACE* def_face[],        // faces being offset
    double def_offset,       // default distance
                             // to offset
    int const& n_spec_face,  // number of faces with
                             // specific offsets
    FACE* spec_face[],       // faces with specific
                             // offsets
    double spec_offset[],    // specific offset values
    BODY*& sheet_body,       // resultant output sheet
                             // body
    SPAposition box_low,     // start of intersection
                             // box to be used
    SPAposition box_high,    // end of intersection
                             // box to be used
    AcisOptions* ao          // acis options
        = NULL               //
    );
```

Includes:

```
#include ”kernel/acis.hxx”
#include ”baseutil/vector/position.hxx”
#include ”kernel/kernapi/api/acis_options.hxx”
#include ”kernel/kernapi/api/api.hxx”
#include ”kernel/kerndata/top/body.hxx”
#include ”kernel/kerndata/top/face.hxx”
#include ”shl_husk/api/shl_api.hxx”
```

Description: Offsets surfaces of the supplied faces by an offset distance specific for each face. Creates a new sheet body out of those faces. The sheet body is converted to two–dimensional sheet.

Radial faces with surfaces which cannot be so offset are removed and the resulting wound healed by the surrounding face surfaces. Optionally repair body can be used during the offset.

Topology Changes:

Refer to the topology changes listed for the Scheme extension, lop:offset-specific.

Geometry Changes:

Refer to the geometry changes listed for the Scheme extension, lop:offset-specific.

Errors: Refer to the errors listed for api_offset-specific.

Limitations: Refer to the Limitations listed for api_offset-specific.

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: Changes model

api_sheet_thicken

Function: Shelling

Action: Creates a solid body from a sheet.

Prototype:

```
outcome api_sheet_thicken (
    BODY* body,              // sheet to be thickened
    double thickness,        // distance to offset
    logical both,            // direction(s) to
                             // thicken
    SPAposition box_low,     // start of intersection
                             // box to be used
    SPAposition box_high,    // end of intersection
                             // box to be used
    AcisOptions* ao = NULL   // ACIS options
    );
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/top/body.hxx"
#include "shl_husk/api/shl_api.hxx"
#include "baseutil/vector/position.hxx"
#include "baseutil/logical.h"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description:

The purpose of sheet thickening is to make a realistic three–dimensional sheet from an ideal two–dimensional one. So usually the thickness supplied will be very small.

This API makes a copy of the sheet to be thickened, offsets it, and fits new orthogonal side faces between the two copies. If the keyword both is not supplied and the thickness supplied is positive, then the sheet is thickened in the direction of the normal of the sheet. Supplying a negative thickness thickens it in the opposite direction. If the keyword both is supplied, then the sheet is thickened in each direction by half the thickness.

The optional intersection box limits the size of intersections between surfaces which might otherwise be very long. It can not be used to choose faces. Its main purpose is to speed up complicated cases where the intersection curves might be very long, thus improving performance. The box must contain the final faces and if omitted defaults to the size box.

Topology Changes:

Refer to the topology changes listed for the function, api_tweak_faces.

Vertices may be split as a result of thickening the sheet.

Mergeable edges will be removed unless they have a NO_MERGE_ATTRIB.

Geometry Changes:

Refer to the geometry changes listed for the function, api_tweak_faces.

Errors:

Refer to the errors listed for the function, api_tweak_faces.

Limitations: The sheet must be manifold. In particular that means:

– There cannot be more than two sheet edges meeting at any vertex and the sheet must consist of no more than one shell.

– If the sheet is made up of several faces, the normals of adjacent faces must be consistent.

– Laminas cannot be thickened.

Only single–sided sheets can be thickened. If a double–sided sheet is encountered it is made single sided.

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: Changes model.

api_terminate_shelling

Function: Shelling, Modeler Control

Action: Terminates the shelling library.

Prototype:
```
outcome api_terminate_shelling ();
```

Includes:
```
#include ″kernel/acis.hxx″
#include ″kernel/kernapi/api/api.hxx″
#include ″shl_husk/api/shl_api.hxx″
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: shl_husk

Filename: shl/shl_husk/api/shl_api.hxx

Effect: System routine