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_phlv5_clean

Function: Hidden Line Removal

Action: Removes PHLV5 attributes from a list of bodies.

Prototype:
```
outcome api_phlv5_clean (
    ENTITY_LIST const& bodies,  // body list
    int token,                  // view token
    AcisOptions* ao = NULL      // acis options
    );
```

Includes:
```
#include "phlv5_husk/api/phlv5_api.hxx"
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/lists/lists.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: This API removes and deletes any PHLV5 attributes with a matching view token from the bodies list.

Errors: None

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/api/phlv5_api.hxx

Effect: Changes model

api_phlv5_compute

Function: Hidden Line Removal

Action: Computes hidden line data for the given viewing parameters and list of bodies.

Prototype:
```
outcome api_phlv5_compute (
    ENTITY_LIST const& bodies,  // body list
    int token,                  // view token
    SPAposition const& eyepos,  // eye position
    SPAposition const& target,  // target position
    logical persp,              // perspective
                                // projection flag
    ENTITY_LIST& edges,         // output list of
                                // edges
    phlv5_options*  phlv5_opts = NULL,// option class
    AcisOptions* ao = NULL      // acis options
    );
```

Includes:
```
#include "phlv5_husk/api/phlv5_api.hxx"
#include "phlv5_husk/sg_husk/phlv5/phlv5_opts.hxx"
#include "baseutil/vector/position.hxx"
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/lists/lists.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: If the view token is nonzero, the data is stored on the model as attributes and identified by token. Make token zero if you do not wish to store the view. Existing attributes identified by the same token are replaced. If the perspective projection flag persp is TRUE, data is calculated for display in a perspective projection; otherwise, calculations are done for a parallel projection.

The options argument is of type phlv5_options and contains options for the hidden line removal algorithm.

Note *All data returned is allocated with* new. *It is the user's responsibility to call* lose *for all the entities in the edge list when they are no longer needed.*

Errors: The distance between eyepos and target is less than SPAresabs.

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/api/phlv5_api.hxx

Effect: Changes model

api_phlv5_retrieve

Function: Hidden Line Removal

Action: Retrieves data stored in PHLV5 attributes with a matching view token and returns it as an ENTITY_LIST of PHLV5_EDGEs.

Prototype:
```
outcome api_phlv5_retrieve (
    ENTITY_LIST const& bodies,  // body list
    int token,                  // view token
    ENTITY_LIST&  hed_list,     // edge list
    PHLV5_CAMERA*& camera,      // camera definition
    AcisOptions* ao = NULL      // acis options
    );
```

Includes:
```
#include "phlv5_husk/api/phlv5_api.hxx"
#include "phlv5_husk/sg_husk/phlv5/phlv5_cam.hxx"
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/lists/lists.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Retrieves data stored in PHLV5 attributes with a matching view token and returns it as an ENTITY_LIST of PHLV5_EDGEs. The attributes are placed on the body by api_phlv5_compute when the api_phlv5_compute's token is set to TRUE.

This API also returns the camera definition.

Note *All data returned is a copy of the data stored in attributes on the body. It is the user's responsibility to call* lose *for the returned camera and all entities in the list, when they are no longer needed.*

Errors: The bodies in the list did not have matching PHLV5_CAMERAs.

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/api/phlv5_api.hxx

Effect: Changes model

api_initialize_hidden_line_removal

Function: Hidden Line Removal

Action: Initializes the hidden line library.

Prototype:
```
outcome api_initialize_hidden_line_removal ();
```

Includes:
```
#include ”kernel/acis.hxx”
#include ”phlv5_husk/api/phlv5_api.hxx”
#include ”kernel/kernapi/api/api.hxx”
```

Description: Initializes the hidden line library.

Errors: None

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/api/phlv5_api.hxx

Effect: System routine

api_terminate_hidden_line_removal

Function: Hidden Line Removal

Action: Terminates the hidden line library.

Prototype:
```
outcome api_terminate_hidden_line_removal ();
```

Includes:
```
#include ”kernel/acis.hxx”
#include ”phlv5_husk/api/phlv5_api.hxx”
#include ”kernel/kernapi/api/api.hxx”
```

Description: Terminates the hidden line library.

Errors: None

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/api/phlv5_api.hxx

Effect: System routine

is_PHLV5_EDGE

Function: Object Relationships

Action: Determines if the entity is a PHLV5_EDGE.

Prototype:
```
logical is_PHLV5_EDGE (
    const ENTITY* e          // Entity to test
    );
```

Includes:
```
#include ”kernel/acis.hxx”
#include ”baseutil/logical.h”
#include ”kernel/geomhusk/acistype.hxx”
#include ”phlv5_husk/sg_husk/phlv5/phlv5_edge.hxx”
```

Description: Determines if the entity is a PHLV5_EDGE.

Errors: None

Limitations: None

Library: phlv5_husk

Filename: phlv5/phlv5_husk/sg_husk/phlv5/phlv5_edge.hxx

Effect: System routine

is_PHLV5_SEGMENT

Function: Object Relationships

Action: Determines if the entity is a PHLV5_SEGMENT.

Prototype:
```
logical is_PHLV5_SEGMENT (
    const ENTITY* e          // Entity to test
    );
```

Includes:
```
#include ”kernel/acis.hxx”
#include ”baseutil/logical.h”
#include ”kernel/geomhusk/acistype.hxx”
#include ”phlv5_husk/sg_husk/phlv5/phlv5_seg.hxx”
```

Description:	Determines if the entity is a PHLV5_SEGMENT.
Errors:	None
Limitations:	None
Library:	phlv5_husk
Filename:	phlv5/phlv5_husk/sg_husk/phlv5/phlv5_seg.hxx
Effect:	System routine