

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_ihl_clean

Function: Interactive Hidden Line

Action: Removes interactive hidden line data attributes from the model.

Prototype:

```
outcome api_ihl_clean (
    const ENTITY_LIST& entities, // entity list
    int token                  // view token
);
```

Includes:

```
#include "kernel/acis.hxx"
#include "ihl_husk/api/ihlapi.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/lists/lists.hxx"
```

Description: This API removes and deletes any IHL attributes with a matching view token from the list of entities.

When entities are cleaned, attributes are removed and are not stored. These attributes cannot be retrieved using the `api_ihl_retrieve` function.

The optional token is an integer index. When token is 0, no action is taken. A nonzero token causes hidden line data to be removed from attributes attached to the entities for the view matching that view token; attributes for other views are not affected.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/api/ihlapi.hxx

Effect: Changes model

api_ihl_compute

Function: Interactive Hidden Line

Action: Computes interactive hidden line data and optionally stores it on the model as attributes.

Prototype:

```
outcome api_ihl_compute (  
    const ENTITY_LIST& entities, // entity list  
    int token,                 // view token  
    const SPAPosition& eyepos,  // view eye  
    const SPAPosition& target,  // view target  
    logical fPersp,             // TRUE if  
                                // perspective  
                                // projection  
    logical fIntrSeg,           // TRUE if interior  
                                // segments needed  
    logical fHidSeg,           // TRUE if hidden  
                                // segments needed  
    logical unfacetted,        // TRUE to avoid  
                                // refaceting entity  
    logical fNoHidCalc,        // TRUE if you do not  
                                // want to calculate  
                                // hidden line but  
                                // only silhouette  
    ENTITY_LIST& segments      // segments returned  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "ihl_husk/api/ihlapi.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "kernel/kerndata/lists/lists.hxx"  
#include "baseutil/logical.h"  
#include "baseutil/vector/position.hxx"
```

Description: This API computes the hidden line data for the given viewing parameters and list of entities. The list of entities is run through the faceter to generate a corresponding list of meshes. The meshes are then used to generate hidden line data. The hidden line data consists of 2D line segments stored in the ENTITY_LIST segments.

If the view token is nonzero, the data is stored on the model as ATTRIB_IHL_VW attributes and identified by token. Existing attributes identified by the same token are removed. If the perspective projection flag fPersp is TRUE, data is calculated for display in a perspective projection; otherwise, calculations are done for a parallel projection.

All data returned is allocated with new. 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 distance between eyepos and target is less than SPAsabs.

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/api/ihlapi.hxx

Effect: Changes model

api_ihl_compute_entity_silhouette

Function: Interactive Hidden Line

Action: Computes silhouette segments from interactive hidden line data.

Prototype:

```
outcome api_ihl_compute_entity_silhouette (  
    const ENTITY_LIST&          // list of entities  
    entities,                   // to compute from  
    const SPAPosition& eyepos,  // eye position  
    const SPAPosition& target,  // target position  
    logical fPersp,             // TRUE if perspective  
    logical fHidSeg,            // TRUE if hidden  
                                // segments needed  
    logical unfacetted,         // TRUE to avoid  
                                // refaceting entity  
    logical fNoHidCalc,         // TRUE to avoid hidden  
                                // line calculation  
    ENTITY_LIST& segments      // segments returned  
);
```

Includes:	<pre>#include "kernel/acis.hxx" #include "ihl_husk/api/ihlapi.hxx" #include "kernel/kernapi/api/api.hxx" #include "kernel/kerndata/lists/lists.hxx" #include "baseutil/logical.h" #include "baseutil/vector/position.hxx"</pre>
Description:	This API computes the faceted silhouette segment data for the given viewing parameters and list of entities. The list of entities is processed and then is used to generate faceted silhouette segment data. The hidden line data consists of 3D line segments stored in the ENTITY_LIST segments. If the perspective projection flag fPersp is TRUE, data is calculated for display in a perspective projection; otherwise, calculations are done for a parallel projection. All data returned is allocated with new. It is the user's responsibility to handle them when they are no longer needed.
Errors:	An error results if the distance between the eye position and the target is less than SPAsesabs.
Limitations:	None
Library:	ihl_husk
Filename:	ihl/ihl_husk/api/ihlapi.hxx
Effect:	Changes model

api_ihl_compute_from_meshes

Function: Interactive Hidden Line

Action: Creates interactive hidden line output from a set of meshes.

Prototype:

```
outcome api_ihl_compute_from_meshes (
    POLYEDGE_MESH* meshes[], // list of meshes
    TRANSFORM* transfs[],    // list of transforms
    int num_meshs,           // number of meshes and
                             // transforms in list
    const SPAPosition& eyepos, // eye position
    const SPAPosition& target, // target position
    logical fPersp            // TRUE if perspective
                             // projection
);
```

Includes:	<pre>#include "kernel/acis.hxx" #include "ihl_husk/api/ihlapi.hxx" #include "ihl_husk/meshmgr/pe_mesh.hxx" #include "kernel/kernapi/api/api.hxx" #include "kernel/kerndata/geom/transfrm.hxx" #include "baseutil/logical.h" #include "baseutil/vector/position.hxx"</pre>
Description:	Uses the current output manager to generate hidden line data. An output manager of the IHL_OUTPUT_MANAGER class must be made current using the api_ihl_set_output_manager function before this function is called. Normally hidden line data is created using api_ihl_compute, which accepts a list of bodies, generates a list of corresponding meshes using the faceter, and calculates hidden line data from these meshes. If the list of meshes already exists (i.e., the faceter does not need to be run again), api_ihl_compute_from_meshes can be called to generate hidden line data directly. If the perspective projection flag fPersp is TRUE, data is calculated for display in a perspective projection; otherwise, calculations are done for a parallel projection. All data returned is allocated with new. It is the user's responsibility to call lose for the camera and all entities in the list when they are no longer needed.
Errors:	None
Limitations:	None
Library:	ihl_husk
Filename:	ihl/ihl_husk/api/ihlapi.hxx
Effect:	Changes model

api_ihl_facet

Function:	Interactive Hidden Line
Action:	Facets the given entities and attaches the resulting POLYEDGE_MESH to each entity using an attribute.

Prototype: `outcome api_ihl_facet (`
 `const ENTITY_LIST& entities, // entities to facet`
 `logical unfacetted                // if TRUE, do not`
 `// refacet if already`
 `// faceted`
 `);`

Includes: `#include "kernel/acis.hxx"`
 `#include "baseutil/logical.h"`
 `#include "ihl_husk/api/ihlapi.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 `#include "kernel/kerndata/lists/lists.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/api/ihlapi.hxx

Effect: Changes model

api_ihl_get_mesh

Function: Interactive Hidden Line

Action: Gets a POLYEDGE_MESH for an entity, faceting the entity if necessary first.

Prototype: `outcome api_ihl_get_mesh (`
 `ENTITY* entity,                // entity list`
 `POLYEDGE_MESH*& mesh        // returned mesh`
 `);`

Includes: `#include "kernel/acis.hxx"`
 `#include "ihl_husk/api/ihlapi.hxx"`
 `#include "ihl_husk/meshmgr/pe_mesh.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 `#include "kernel/kerndata/data/entity.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: ihl_husk
Filename: ihl/ihl_husk/api/ihlapi.hxx
Effect: Changes model

api_ihl_get_output_manager

Function: Interactive Hidden Line
Action: Gets the current output manager.
Prototype:

```
outcome api_ihl_get_output_manager (  
    IHL_OUTPUT_MANAGER*& ihlOM // current output  
                                // manager  
);
```


Includes:

```
#include "kernel/acis.hxx"  
#include "ihl_husk/api/ihlapi.hxx"  
#include "ihl_husk/meshmgr/ihloutp.hxx"  
#include "kernel/kernapi/api/api.hxx"
```


Description: The output manager returned is used internally to output hidden line data.
Errors: None
Limitations: None
Library: ihl_husk
Filename: ihl/ihl_husk/api/ihlapi.hxx
Effect: Read-only

api_ihl_occlusion

Function: Interactive Hidden Line
Action: Removes overlapping invisible segments from the IHL segment list.
Prototype:

```
outcome api_ihl_occlusion (  
    ENTITY_LIST& ihl_segs, // ihl segments  
    double tol             // tolerance value  
        = SPAsesfit        // default is SPAsesfit  
);
```


Includes:

```
#include "kernel/acis.hxx"  
#include "ihl_husk/api/ihlapi.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "kernel/kerndata/lists/lists.hxx"
```

Description:	This API eliminates overlapping invisible segments. A tolerance value is used to determine overlapping conditions. Visible segments are not affected.
Errors:	None
Limitations:	None
Library:	ihl_husk
Filename:	ihl/ihl_husk/api/ihlapi.hxx
Effect:	Changes model

api_ihl_retrieve

Function: Interactive Hidden Line

Action: Retrieves interactive hidden line data from attributes attached to the entities (faces or bodies).

Prototype:

```
outcome api_ihl_retrieve (
    const ENTITY_LIST& entities, // body list
    int token,                  // view token
    ENTITY_LIST& segments,      // segment list
    IHL_CAMERA*& camera         // camera
);
```

Includes:

```
#include "kernel/acis.hxx"
#include "ihl_husk/api/ihlapi.hxx"
#include "ihl_husk/ihl/ihl_cam.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/lists/lists.hxx"
```

Description: This API retrieves the data stored in attributes with a matching view token and returns it as an ENTITY_LIST of segments. The attributes are placed on the entity by api_ihl_compute when the api_ihl_compute's token is set to TRUE. Attributes remain attached to entities.

This API also returns the camera definition.

All data returned is a copy of the data stored in attributes on the entity. 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 entities in the list did not have matching IHL_CAMERA's.

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/api/ihlapi.hxx

Effect: Read-only

api_ihl_set_output_manager

Function: Interactive Hidden Line

Action: Sets the current output manager.

Prototype:

```
outcome api_ihl_set_output_manager (  
    IHL_OUTPUT_MANAGER* ihlOM    // new output manager  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "ihl_husk/api/ihlapi.hxx"  
#include "ihl_husk/meshmgr/ihloutp.hxx"  
#include "kernel/kernapi/api/api.hxx"
```

Description: Needs to be called only prior to calling `api_ihl_compute_from_meshes`. The output manager gives an application full control over how hidden line data is output.

Generally an application is expected to derive an application-specific class from the `IHL_OUTPUT_MANAGER`, and then pass this output manager to `api_ihl_set_output_manager`.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/api/ihlapi.hxx

Effect: System routine

api_initialize_interactive_hidden_line

Function: Interactive Hidden Line, Modeler Control

Action: Initializes the interactive hidden line library.

Prototype: `outcome api_initialize_interactive_hidden_line ();`
 Includes: `#include "kernel/acis.hxx"`
 `#include "ihl_husk/api/ihlapi.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 Description: Refer to Action.
 Errors: None
 Limitations: None
 Library: `ihl_husk`
 Filename: `ihl/ihl_husk/api/ihlapi.hxx`
 Effect: System routine

api_terminate_interactive_hidden_line

Function: Interactive Hidden Line, Modeler Control
 Action: Terminates the interactive hidden line library.
 Prototype: `outcome api_terminate_interactive_hidden_line ();`
 Includes: `#include "kernel/acis.hxx"`
 `#include "ihl_husk/api/ihlapi.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 Description: Refer to Action.
 Errors: None
 Limitations: None
 Library: `ihl_husk`
 Filename: `ihl/ihl_husk/api/ihlapi.hxx`
 Effect: System routine

is_ENTITY_IHL

Function: Interactive Hidden Line
 Action: Determines if an ENTITY is an ENTITY_IHL
 Prototype: `logical is_ENTITY_IHL (`
 `const ENTITY* e                   // entity to test`
 `);`

Includes: #include "kernel/acis.hxx"
 #include "baseutil/logical.h"
 #include "ihl_husk/ihl/en_ihl.hxx"
 #include "kernel/kerndata/data/entity.hxx"

Description: Refer to Action.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/ihl/en_ihl.hxx

Effect: Read-only

is_IHL_CAMERA

Function: Interactive Hidden Line

Action: Determines if an ENTITY is an IHL_CAMERA.

Prototype: logical is_IHL_CAMERA (
 const ENTITY* e // entity to test
);

Includes: #include "kernel/acis.hxx"
 #include "baseutil/logical.h"
 #include "ihl_husk/ihl/ihl_cam.hxx"
 #include "kernel/kerndata/data/entity.hxx"

Description: Refer to Action.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/ihl/ihl_cam.hxx

Effect: Read-only

is_IHL_SEGMENT

Function: Interactive Hidden Line

Action: Determines if an ENTITY is an IHL_SEGMENT.

Prototype: logical is_IHL_SEGMENT (
 const ENTITY* e // entity to test
);

Includes: `#include "kernel/acis.hxx"`
 `#include "baseutil/logical.h"`
 `#include "ihl_husk/ihl/ihl_seg.hxx"`
 `#include "kernel/kerndata/data/entity.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: ihl_husk

Filename: ihl/ihl_husk/ihl/ihl_seg.hxx

Effect: Read-only