

Chapter 7.

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_advanced_cover

Function: Covering, Model Topology, Creating Entities

Action: Calculates a new surface covering a circuit of edges and adds it to body.

Prototype:

```
outcome api_advanced_cover (
    FACE* target_face,          // FACE to be re-covered
    acover_options* options    // Advanced Covering
        = NULL,                // options
    AcisOptions* opts          // general ACIS options
        = NULL                 //
);

outcome api_advanced_cover (
    FACE*&new_face,            // FACE created
    EDGE* edge_in_circuit,     // edge in the circuit to
                                // be covered
    acover_options* options    // Advanced Covering
        = NULL,                // options
    AcisOptions* opts          // general ACIS options
        = NULL                 //
);
```

Includes:	<pre>#include "kernel/acis.hxx" #include "admsk/acover/acover_opts.hxx" #include "admsk/acover/acovrapi.hxx" #include "kernel/kernapi/api/acis_options.hxx" #include "kernel/kernapi/api/api.hxx" #include "kernel/kerndata/top/edge.hxx" #include "kernel/kerndata/top/face.hxx"</pre>
Description:	This API creates a face by covering the specified circuit of edges. The circuit can surround either a hole (first signature, taking an EDGE* and returning a FACE*) or a FACE that is already a part of the model (second signature, taking a FACE*). When using the first signature, ACIS uses the input edge to find a circuit of unshared (i.e. sheet boundary or wire) edges that include that edge. When using the second signature, the circuit of edges is just the set of edges corresponding to the outer loop of the face. After finding a circuit, ACIS calculates a covering surface which embeds the edges in the circuit and adds it to the model. When using the second signature, the surface simply replaces the geometry of the existing FACE argument (target_face). When using the first signature, ACIS creates a new FACE and hooks it into the model (see api_cover_circuits). This face is returned in the "new_face" argument. Note that new_face must be set to NULL before passing it in to api_advanced_cover; this is to protect users from accidentally overwriting active face pointers. After the new face has been added to the model, edges in the boundary circuit are made tolerant as necessary to avoid gap errors. The advance covering options are used provide additional information to be used in creating the surface, such as continuity constraints on the boundary edges, shape parameters, or number of B-Spline spans to be used in the covering surface.
Errors:	None
Limitations:	See Advanced Covering theory chapter for a discussion of limitations.
Library:	admsk
Filename:	adm/admsk/acover/acovrapi.hxx
Effect:	Changes model

api_dm_add_multi_face

Function:	Deformable Surfaces
Action:	Adds a face to the deformable modeling multi-surface structure.

Prototype:

```

outcome api_dm_add_multi_face (
    ATTRIB_DM2ACIS*      // sculpting data
        attrib_dm2acis,  // attribute
    ENTITY* entity,      // FACE or EDGE to
                        // connect
    int& entity_tag,      // tag number for dmod
    int& tag_shift,       // amount face's tag
                        // numbers have shifted
                        // new_tag_id =
                        // old_tag_id + tag_shift
    adm_options* adm_o    // adm options
    = NULL,               //
    AcisOptions* acisO    // ACIS options such as
    = NULL                // version and journal
);

```

Includes:

```

#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "kernel/kerndata/data/entity.hxx"
#include "admhusk/ds2acis/ds2acis.hxx"
#include "admhusk/ds2acis/dsapi.hxx"
#include "admhusk/ds2acis/adm_options.hxx"
#include "kernel/kernapi/api/acis_options.hxx"

```

Description: Attaches attributes to the input face that includes it in the deformable modeling set described by the ATTRIB_DM2ACIS.

This function will allow any set of faces to be connected together. However, only faces with compatible parameterizations can be deformed without introducing position and tangent tolerance gaps between the edges. Boundaries between two compatibly parameterized faces may be deformed as one while the boundary stays C1 to within numerical accuracy (SPAresabs).

If the input face is an existing deformable model, all of its tag numbers, for deformable models, loads, and constraints, will be changed to make sure that they are unique in the entire multi-surface deformable modeling hierarchy. The number of changed tag numbers is returned in `adj_tag_count`. When `adj_tag_count` is larger than 0, the `adj_tag` argument is set to point to an allocated array containing an old-tag-number, new-tag-number pair for every changed tag number organized as, [old-tag1, new-tag1, old-tag2, new-tag2, ...]. The size of the `adj_tag` array is [2*`adj_tag_count`]. The calling function must free the `adj_tag` array when done with it.

Errors: DS_API_NO_ATTRIB_DM2ACIS
The input DS_API_NO_ATTRIB_DM2ACIS value cannot be NULL.

DS_API_NOT_FACE_OR_EDGE
The entity is not a face or an edge, and therefore is not sculptable.

DS_NULL_INPUT_PTR
The face cannot be NULL on input.

Limitations: Currently only quad faces can be combined to make a multi-surface deformable model. We do not yet support a multi-edge network. We support a quilt of trimmed patches.

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_add_patch

Function: Deformable Surfaces

Action: Adds a deformable patch to an entity for local deformations.

```

Prototype:  outcome api_dm_add_patch (
            ATTRIB_DM2ACIS*          // sculpting data
            attrib_dm2acis,          // attribute
            int shape_flag,           // 1 = add rectangle,
                                     // 2 = add ellipse,
                                     // 3 = add fillet_square
            SPapar_pos domain_pt1,    // rect = min-corner,
                                     // ellipse = center-pt
            SPapar_pos domain_pt2,    // rect = max-corner,
                                     // ellipse = 1st-axis-
                                     // end-pt
            SPapar_pos domain_pt3,    // rect = not-used,
                                     // ellipse = 2nd-axis-
                                     // end-pt
            int refinement,           // parent_knot_spacing/
                                     // child_knot_spacing
            int tag_flag,             // specify parent
                                     // tgt_dmod to
                                     // get new patch
            int& patch_tag,           // specify parent
                                     // tgt_dmod to
                                     // get new patch
            adm_options* adm_o        // adm options
            = NULL,                  //
            AcisOptions* acisO        // ACIS options such as
            = NULL                    // version and journal
        );

```

```

Includes:  #include "kernel/acis.hxx"
            #include "kernel/kernapi/api/api.hxx"
            #include "baseutil/vector/param.hxx"
            #include "admhusk/ds2acis/ds2acis.hxx"
            #include "admhusk/ds2acis/dsapi.hxx"
            #include "admhusk/ds2acis/adm_options.hxx"
            #include "kernel/kernapi/api/acis_options.hxx"

```

Description: Adds a patch to a parent target deformable model in preparation for local deformations. The `shape_flag` argument specifies the shape of the new patch. When `shape_flag` equals 1, a rectangular patch is added whose lower corner is `domain_pt1` and whose upper corner is `domain_pt2`. When `shape_flag` equals 2, an elliptical patch is added whose center point is `domain_pt1`, the first axis is the vector from `domain_pt1` to `domain_pt2`, and the second axis is the vector from `domain_pt1` to `domain_pt3`. When `shape_flag` equals 3, a rectangular patch is added whose corners are given arc fillets of radius equal to `domain_pt3`.

When `shape_flag` equals 1, `domain_pt3` is not used. The domain point arguments are all `SPAPar_pos` objects. They specify locations in the domain space of the target parent deformable model whose domain space always maps to the unit square.

The `refinement` argument specifies the density of the patch control points as compared to the parents. A `refinement` value of 1 specifies that the new patch control point density is the same as that of the target deformable model. A `refinement` value of 2 specifies that the new patch has twice the density of control points as the parent. The `refinement` value must be an integer greater than 0.

The `tag_flag` argument specifies which deformable model within the `attrib_dm2acis` patch hierarchy is to be the parent of the new patch. When `tag_flag` equals 1 the active deformable model receives the patch. When `tag_flag` equals 2 the root deformable model receives the patch. Any other value specifies the deformable model whose tag value is equal to the `tag_flag`.

The `target` argument specifies which deformable model to use in a patch hierarchy. Valid values for `target` are:

- 1 = active deformable model
- 2 = root deformable model
- 1 = active deformable model and offspring
- 2 = root deformable model and offspring

Otherwise, the `target` is the deformable model whose tag identifier equals `target`.

Errors:

`DS_API_NO_ATTRIB_DM2ACIS`

The input `DS_API_NO_ATTRIB_DM2ACIS` value cannot be NULL.

`DS_API_NOT_FACE_OR_EDGE`

The entity is not a face or an edge, and therefore is not sculptable.

`DS_API_NOT_IN_HIERARCHY`

The value of the target deformable model is not valid.

`DS_API_BAD_PATCH_DOMAIN`

The patch domain is not contained by a parent or it overlaps a sibling patch.

Limitations:

Elliptical patches may not be added. Face patches must be completely contained within the interior of their parents; the boundary of the child may not touch or overlap the boundary of its parent.

Library: admhusk
Filename: adm/admhusk/ds2acis/dsapi.hxx
Effect: Changes model

api_dm_auto_elevate

Function: Deformable Surfaces, Modeler Control

Action: Sets whether curves or surfaces with degree less than 3 are automatically elevated to degree 3 when brought into deformable modeling.

Prototype:

```
outcome api_dm_auto_elevate (  
    int y_n,                // autoelevate key  
    adm_options* adm_o      // adm options  
    = NULL,                 //  
    AcisOptions* acis0      // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: By default, curves or surfaces with degree less than 3 are automatically elevated to degree 3 when brought into deformable modeling. This API allows a user to toggle this behavior. Calling `api_dm_auto_elevate` will turn off auto-elevate for all curves and surfaces subsequently brought into deformable modeling. Calling with a non-zero argument will turn on auto-elevate for all curves or surfaces subsequently brought into deformable modeling.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_commit_attrib_dm2acis

Function: Deformable Surfaces

Action: Copies the attribute shape data into its owner's shape representation and copies the deformable model data into its owner's attribute.

Prototype: `outcome api_dm_commit_attrib_dm2acis (`
 `ATTRIB_DM2ACIS*            // pointer to the owner's`
 `attrib_dm2acis,        // deformable model`
 `// attribute`
 `adm_options* adm_o        // adm options`
 `= NULL,                                //`
 `AcisOptions* acis0        // ACIS options such as`
 `= NULL                                // version and journal`
 `);`

Includes: `#include "kernel/acis.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 `#include "admhusk/ds2acis/ds2acis.hxx"`
 `#include "admhusk/ds2acis/dsapi.hxx"`
 `#include "admhusk/ds2acis/adm_options.hxx"`
 `#include "kernel/kernapi/api/acis_options.hxx"`

Description: Replaces the owner's geometry description with the spline description in the ATTRIB_DM2ACIS attribute. Sculpting operations affect only the non-persistent data of the ATTRIB_DM2ACIS attribute. Sculpting changes to the ATTRIB_DM2ACIS attribute can not be undone with rollback. When an end user is satisfied with a sculpted shape, this function is used to commit the sculpting changes to the owning entity. Changes made to the owner are backed up and can be undone with rollback.

Additionally, this function stores the current state of the deformable model within a persistent ATTRIB_DSMODEL attribute.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_get_attrib_dm2acis

Function: Deformable Surfaces

Action: Gets an ATTRIB_DM2ACIS attribute for the input entity if one exists or creates a new one.

Prototype: outcome api_dm_get_attrib_dm2acis (
 ENTITY* entity, // in:target FACE or EDGE
 ATTRIB_DM2ACIS*& // out: pointer to
 attrib_dm2acis, // new or retrieved
 // attrib_dm2acis for
 // entity
 adm_options* adm_o // adm options
 = NULL, //
 AcisOptions* acisO // ACIS options such as
 = NULL // version and journal
);

Includes: #include "kernel/acis.hxx"
 #include "kernel/kernapi/api/api.hxx"
 #include "kernel/kerndata/data/entity.hxx"
 #include "admhusk/ds2acis/ds2acis.hxx"
 #include "admhusk/ds2acis/dsapi.hxx"
 #include "admhusk/ds2acis/adm_options.hxx"
 #include "kernel/kernapi/api/acis_options.hxx"

Description: Searches the entity's patch hierarchy for an ATTRIB_DM2ACIS and
 returns a pointer to it when found, after ensuring that the
 ATTRIB_DM2ACIS internal data structures are up to date.

When no ATTRIB_DM2ACIS is found a new attribute is created for the
 entity using either ATTRIB_DSMODEL or default values and a pointer to
 the new ATTRIB_DM2ACIS is returned. When the entity is not part of a
 patch hierarchy a one member ATTRIB_DSGROUP and an
 ATTRIB_DSMODEL with this entity are created as the root and are added
 to the entity.

Errors: DS_API_NOT_FACE_OR_EDGE
 The entity is not a face or an edge, and therefore is not sculptable.

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_get_hierarchy_entities

Function: Deformable Surfaces

Action: Sizes the DM_dbl_array and initializes the memory.

Prototype:

```
outcome api_dm_get_hierarchy_entities (
    ATTRIB_DM2ACIS* attrib_dm2acis, // in: ADM
                                   // attribute
    ENTITY_LIST& elist,           // out: entity list
    adm_options* adm_o           // adm options
    = NULL,                      //
    AcisOptions* acisO           // ACIS options such as
    = NULL                      // version and journal
);
```

Includes:

```
#include "admhusk/ds2acis/dsapi.hxx"
#include "kernel/kerndata/lists/lists.hxx"
#include "admhusk/ds2acis/ds2acis.hxx"
#include "kernel/acis.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "admhusk/ds2acis/adm_options.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: In ACIS deformable modeling, every deformable model corresponds to one or more ACIS ENTITIES, either FACES or EDGES. This query returns the ENTITYs associated with all the DS_dmod's in the deformable modeling hierarchy.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: System routine

api_dm_journal_off

Function: Deformable Surfaces

Action: Turns off deformable modeling API level journaling and closes any journal file opened for write by api_dm_journal_on.

Prototype:

```
outcome api_dm_journal_off (
    adm_options* adm_o           // adm options
    = NULL,                      //
    AcisOptions* acisO           // ACIS options such as
    = NULL                      // version and journal
);
```

Includes:	<pre>#include "kernel/acis.hxx" #include "kernel/kernapi/api/api.hxx" #include "admsk/ds2acis/dsapi.hxx" #include "admsk/ds2acis/adm_options.hxx" #include "kernel/kernapi/api/acis_options.hxx"</pre>
Description:	This API closes any file opened by a <code>api_dm_journal_on</code> call and turns off deformable modeling API level journaling. Deformable modeling API journaling is not a modeling tool; it is a debugging tool developed for customers running the deformable modeling API without ACIS or Scheme AIDE.
Errors:	None
Limitations:	None
Library:	admsk
Filename:	adm/admsk/ds2acis/dsapi.hxx
Effect:	System routine

api_dm_journal_on

Function: [Deformable Surfaces](#)

Action:	Enables deformable modeling API level journaling and opens a journal file for write.
Prototype:	<pre>outcome api_dm_journal_on (char* filename, // file to open for write int cascade // 1=log nested API calls // 0=log only entry API // calls adm_options* adm_o // adm options = NULL, // AcisOptions* acis0 // ACIS options such as // version and journal);</pre>
Includes:	<pre>#include "kernel/acis.hxx" #include "kernel/kernapi/api/api.hxx" #include "admsk/ds2acis/dsapi.hxx" #include "admsk/ds2acis/adm_options.hxx" #include "kernel/kernapi/api/acis_options.hxx"</pre>
Description:	Modifies library global values <code>DM_journal_file</code> , <code>DM_journal</code> , and <code>DM_cascade</code> .

A call to this function begins a deformable modeling API journal session. The call opens for write the file indicated by the input filename and sets the global variables DM_journal and DM_cascade to begin journaling all future deformable modeling API calls. Deformable modeling API call journaling is a debugging tool for deformable modeling customers who do not run ACIS or Scheme AIDE. When DM_journal is set to 1, every call to a deformable modeling API routine generates an input report listing all the input variable values when the routine was entered. The deformable modeling API routines also generate an output report listing all the output variable values just before the routine exits.

The cascade argument controls how much information gets written to the journal file. When the value of cascade is:

- 0 = all entry level deformable modeling calls are journaled.
- 1 = all entry level and cascaded deformable modeling calls are journaled.
- 2 = all entry level and intersection callback functions are journaled.
- 3 = all entry level deformable modeling calls, cascaded deformable modeling calls, and intersection callback functions are journaled.

When this function is called with journaling is enabled, it closes the current journal file and opens a new one. Opening a journal file with the same name as the current journal file will cause an overwrite of the old journal file data.

Errors:	DM_NULL_INPUT_PTR The filename is NULL on input
	DM_BAD_CASCADE_VALUE The value of cascade must be 0 to 3
	DM_FAILED_FILE_OPEN_WRITE The designated file is not opened for write
Limitations:	None
Library:	admhusk
Filename:	adm/admhusk/ds2acis/dsapi.hxx
Effect:	System routine

api_dm_query_attrib_dm2acis

Function: [Deformable Surfaces](#)

Action: Gets an ATTRIB_DM2ACIS attribute for the input DS_dmod if one exists or returns NULL.

Prototype:

```
outcome api_dm_query_attrib_dm2acis (  
    DS_dmod* dmod,           // target  
    ATTRIB_DM2ACIS*& dm2acis, // pointer returned  
    adm_options* adm_o       // adm options  
    = NULL,                  //  
    AcisOptions* acis0       // ACIS options such as  
    = NULL                    // version and journal  
);  
  
outcome api_dm_query_attrib_dm2acis (  
    ENTITY* entity,          // target  
    ATTRIB_DM2ACIS*& dm2acis, // pointer returned  
    adm_options* adm_o       // adm options  
    = NULL,                  //  
    AcisOptions* acis0       // ACIS options such as  
    = NULL                    // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "admhusk/ds2acis/ds2acis.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "dshusk/dskernel/dsdmod.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "kernel/kerndata/data/entity.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: This function searches the entity's patch hierarchy for an ATTRIB_DM2ACIS and returns a pointer to it when found. When no ATTRIB_DM2ACIS is found NULL is returned. This API is used by the admicon library to translate an SDM tag id to an ADM tag id.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Read-only

api_dm_remove_attrib_dm2acis

Function: Deformable Surfaces

Action: Removes the non-persistent ATTRIB_DM2ACIS sculpting attribute from its owning entity, either a face or an edge.

Prototype:

```
outcome api_dm_remove_attrib_dm2acis (  
    ATTRIB_DM2ACIS* attrib_dm2acis, // pointer to the  
                                     // owner's  
                                     // deformable  
                                     // model  
                                     // attribute  
    adm_options* adm_o           // adm options  
    = NULL,                      //  
    AcisOptions* acisO           // ACIS options such as  
    = NULL                       // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/ds2acis.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Removes the non-persistent ATTRIB_DM2ACIS attribute from its owning entity without storing the current sculpted shape and deformable model data back within the owning entity. This ends the sculpting session.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_remove_dm_attributes

Function: Deformable Surfaces

Action: Removes all deformable modeling attributes from an entity.

Prototype:

```
outcome api_dm_remove_dm_attributes (  
    ENTITY* entity,           // pointer to entity  
    adm_options* adm_o       // adm options  
    = NULL,                  //  
    AcisOptions* acisO       // ACIS options such as  
    = NULL                    // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "kernel/kerndata/data/entity.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: This API removes all deformable modeling attributes from an entity. It is intended to cause bodies to “forget” that their geometry was created by deformable modeling.

Errors: If the input is not a pointer to a valid entity, system error NULL_ENTITY results.

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_rm_multi_face

Function: Deformable Surfaces

Action: Separates a face from the deformable modeling multi-surface structure and marks it as a stand-alone deformable model. The face is not deleted.

Prototype: outcome api_dm_rm_multi_face (
 ATTRIB_DM2ACIS* // sculpting data
 attrib_dm2acis, // attribute
 int tag, // face to add to
 // deformable set
 adm_options* adm_o // adm options
 = NULL, //
 AcisOptions* acisO // ACIS options such as
 // version and journal
);

Includes: #include "kernel/acis.hxx"
 #include "kernel/kernapi/api/api.hxx"
 #include "admhusk/ds2acis/ds2acis.hxx"
 #include "admhusk/ds2acis/dsapi.hxx"
 #include "admhusk/ds2acis/adm_options.hxx"
 #include "kernel/kernapi/api/acis_options.hxx"

Description: Removes the attributes from the target deformable model's FACE that
 mark it as a member of the multi-surface deformable modeling hierarchy
 modeled by attrib_dm2acis. The target deformable model's FACE is left
 with attributes that mark it as its own standalone deformable modeling
 hierarchy preserving all of its current tag information.

Errors: DS_API_NO_ATTRIB_DM2ACIS
 The input attrib_dm2acis cannot be NULL

 DS_API_NOT_IN_HIERARCHY
 The input tag does not identify a deformable model within the
 attrib_dm2acis hierarchy.

 DS_API_NOT_A_ROOT_SIBLING
 The input tag must be a sibling of the root deformable model

Limitations: This function requires (as a precondition) that only queries have been
 called on the deformable modeling hierarchy since the last call to
 api_dm_commit_attrib_dm2acis.

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_rm_patch

Function: Deformable Surfaces
Action: Removes a patch from a deformable model's patch hierarchy.

Prototype: outcome api_dm_rm_patch (

```

        ATTRIB_DM2ACIS* attrib_dm2acis, // sculpting data
                                           // attribute
        int tag_flag,                      // specify parent
                                           // tgt_dmod to
                                           // get new patch
                                           // 1 =
                                           // active_dmod
                                           // else = dmod
                                           // with tag =
                                           // tag_flag
        adm_options* adm_o                // adm options
        = NULL,                          //
        AcisOptions* acisO               // ACIS options such as
        = NULL                           // version and journal
    );

```

Includes: #include "kernel/acis.hxx"

```

#include "kernel/kernapi/api/api.hxx"
#include "admsk/ds2acis/ds2acis.hxx"
#include "admsk/ds2acis/dsapi.hxx"
#include "admsk/ds2acis/adm_options.hxx"
#include "kernel/kernapi/api/acis_options.hxx"

```

Description: Removes and deletes a deformable model patch and its offspring from the target's parent.

The target argument specifies which deformable model to use in a patch hierarchy. Valid values for target are:

- 1 = active deformable model
- 2 = root deformable model
- 1 = active deformable model and offspring
- 2 = root deformable model and offspring

Otherwise, the target is the deformable model whose tag identifier equals target.

Errors: DS_API_NOT_FACE_OR_EDGE

The entity is not a face or an edge, and therefore is not sculptable.

DS_API_NOT_IN_HIERARCHY

The value of the target deformable model is not valid.

DS_API_BAD_PATCH_DOMAIN

The patch domain is not contained by a parent or it overlaps a sibling patch.

Limitations: This function cannot be used to remove a root deformable model shape. It works only on children within the patch hierarchy.

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_set_array_size

Function: Deformable Modeling

Action: Sizes the DM_dbl_array and initializes the memory.

Prototype:

```
outcome api_dm_set_array_size (
    DM_dbl_array& arr,          // array
    int new_size,              // array size
    double init_val            // value stored in
        = 0.0,                // returned memory
    adm_options* adm_o          // adm options
        = NULL,               //
    AcisOptions* acisO          // ACIS options such as
        = NULL                // version and journal
);
```

Includes:

```
#include "kernel/acis.hxx"
#include "admhusk/ds2acis/dsapi.hxx"
#include "dshusk/dskernel/dm_dbl_array.hxx"
#include "kernel/kernapi/api/api.hxx"
#include "admhusk/ds2acis/adm_options.hxx"
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: The DM_dbl_array is a double array container class returned from deformable modeling queries, outcome 0 or an error value. The DM_dbl_array can also serve as a user container class by sizing with the api_dm_set_array_size method.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: System routine

api_dm_set_tolerance

Function: Deformable Surfaces

Action: Sets the deformable surface tolerance value to be that of the current ACIS tolerance value stored in SPAsresabs.

Prototype:

```
outcome api_dm_set_tolerance (  
    adm_options* adm_o      // adm options  
    = NULL,                //  
    AcisOptions* acis0     // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Sets the deformable surface tolerance values to be the same as the current ACIS tolerance values. SDM uses two tolerance values: DS_tolerance to determine when a distance is equivalent to zero and DS_angle_tol to determine when an angle is equal to zero. This function sets:

```
DS_tolerance = SPAsresabs  
DS_angle_tol = SPAsresnor
```

This API is always called by all the other ADM API functions and so will probably not have to be called directly by an application program.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_dm_use_link_cstrns

Function: Deformable Surfaces, Modeler Control

Action: Overrides the use of link loads in favor of link constraints in the current deformable modeling session.

Prototype:

```
outcome api_dm_use_link_cstrns (  
    adm_options* adm_o      // adm options  
    = NULL,                 //  
    AcisOptions* acisO      // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Sets the deformable modeling session to override the use of link loads in favor of link constraints. This will return the user to pre-release 6.2 behavior. This function should *only* be called at the start of the deformable modeling session. This API is for backward compatibility during the transition from link constraints to link loads.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: Changes model

api_initialize_admhoops

Function: Deformable Surfaces, Modeler Control

Action: Initializes the admhoops library.

Prototype:

```
outcome api_initialize_admhoops ();
```

Includes:

```
#include "kernel/acis.hxx"  
#include "admhoops/admhoops_api.hxx"  
#include "kernel/kernapi/api/api.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: Not applicable

Library: admhoops

Filename: adm/admhoops/admhoops_api.hxx

Effect: System routine

api_initialize_admicon

Function: Deformable Surfaces, Modeler Control

Action: Initializes the admicon library.

Prototype:

```
outcome api_initialize_admicon (  
    adm_options* adm_o      // adm options  
    = NULL,                //  
    AcisOptions* acis0     // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admicon/admicon_api.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: admicon

Filename: adm/admicon/admicon_api.hxx

Effect: System routine

api_initialize_deformable_modeling

Function: Deformable Surfaces, Modeler Control

Action: Initializes the deformable modeling library.

Prototype: `outcome api_initialize_deformable_modeling (`
 `adm_options* adm_o        // adm options`
 `= NULL,                        //`
 `AcisOptions* acis0        // ACIS options such as`
 `= NULL                        // version and journal`
 `);`

Includes: `#include "kernel/acis.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`
 `#include "admsk/ds2acis/dsapi.hxx"`
 `#include "admsk/ds2acis/adm_options.hxx"`
 `#include "kernel/kernapi/api/acis_options.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: admsk

Filename: adm/admsk/ds2acis/dsapi.hxx

Effect: System routine

api_terminate_admhoops

Function: Deformable Surfaces, Modeler Control

Action: Terminates the admhoops library.

Prototype: `outcome api_terminate_admhoops ();`

Includes: `#include "kernel/acis.hxx"`
 `#include "admhoops/admhoops_api.hxx"`
 `#include "kernel/kernapi/api/api.hxx"`

Description: Refer to Action.

Errors: None

Limitations: Not applicable

Library: admhoops

Filename: adm/admhoops/admhoops_api.hxx

Effect: System routine

api_terminate_admicon

Function: Deformable Surfaces, Modeler Control

Action: Terminates the admicon library.

Prototype:

```
outcome api_terminate_admicon (  
    adm_options* adm_o      // adm options  
    = NULL,                 //  
    AcisOptions* acis0      // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "admicon/admicon_api.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: admicon

Filename: adm/admicon/admicon_api.hxx

Effect: System routine

api_terminate_deformable_modeling

Function: Deformable Surfaces, Modeler Control

Action: Terminates the deformable modeling library.

Prototype:

```
outcome api_terminate_deformable_modeling (  
    adm_options* adm_o      // adm options  
    = NULL,                 //  
    AcisOptions* acis0      // ACIS options such as  
    = NULL                  // version and journal  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/kernapi/api/api.hxx"  
#include "admhusk/ds2acis/dsapi.hxx"  
#include "admhusk/ds2acis/adm_options.hxx"  
#include "kernel/kernapi/api/acis_options.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/dsapi.hxx

Effect: System routine

is_ATTRIB_AGC

Function: Deformable Surfaces

Action: Determines if an ENTITY is an ATTRIB_AGC.

Prototype:

```
logical is_ATTRIB_AGC (
    const ENTITY* e           // entity to test
);
```

Includes:

```
#include "kernel/acis.hxx"
#include "kernel/kerndata/data/entity.hxx"
#include "admhusk/ds2acis/agcatt.hxx"
#include "dshusk/dskernel/dmexcept.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/agcatt.hxx

Effect: Read-only

is_ATTRIB_DC2ACIS

Function: Deformable Surfaces

Action: Determines if an ENTITY is an ATTRIB_DC2ACIS.

Prototype:

```
logical is_ATTRIB_DC2ACIS (
    const ENTITY* e           // entity to test
);
```


Includes: `#include "kernel/acis.hxx"`
`#include "kernel/kerndata/data/entity.hxx"`
`#include "admhusk/ds2acis/ds2acis.hxx"`
`#include "dshusk/dskernel/dmexcept.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/ds2acis.hxx

Effect: Read-only

is_ATTRIB_DM2ACIS

Function: Deformable Surfaces

Action: Determines if an ENTITY is an ATTRIB_DM2ACIS.

Prototype:

```
logical is_ATTRIB_DM2ACIS (
    const ENTITY* e          // entity to test
);
```

Includes: `#include "kernel/acis.hxx"`
`#include "kernel/kerndata/data/entity.hxx"`
`#include "admhusk/ds2acis/ds2acis.hxx"`
`#include "dshusk/dskernel/dmexcept.hxx"`

Description: Refer to Action.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/ds2acis.hxx

Effect: Read-only

is_ATTRIB_DS2ACIS

Function: Deformable Surfaces

Action: Determines if an ENTITY is an ATTRIB_DS2ACIS.

Prototype: logical is_ATTRIB_DS2ACIS (
 const ENTITY* e // entity to test
);

Includes: #include "kernel/acis.hxx"
 #include "kernel/kerndata/data/entity.hxx"
 #include "admhusk/ds2acis/ds2acis.hxx"
 #include "dshusk/dskernel/dmexcept.hxx"

Description: Refer to Action.

Errors: None

Limitations: None

Library: admhusk

Filename: adm/admhusk/ds2acis/ds2acis.hxx

Effect: Read-only