

UF_SKET_add_conics [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Adds conics to the current sketch.

The following restrictions applies:

- 1) UF_SKET_initialize_sketch must have been previously called.
- 2) The construction coordinate system of the conic you wish to add must be parallel to the sketch plane.
- 3) Object has to be either a hyperbola, parabola, or a partial ellipse with angular span smaller than 180 degree.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

History

This function was originally released in V17.0

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_add_conics
(
    tag_t sketch_tag,
    int count,
    tag_t * object
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch tag.
<code>int</code>	<code>count</code>	Input	Number of conics to be added.
<code>tag_t *</code>	<code>object</code>	Input	Array of conics to be added to the sketch

UF_SKET_add_extracted_objects [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This routine adds extracted objects to a sketch. New objects, extracted from the input objects and become associative with the input objects, will be added to the sketch.

The following restrictions applies:

UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and external.

See Also

[UF_SKET_initialize_sketch](#)

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_add_extracted_objects
(
    tag_t sketch_tag,
    int count,
    tag_t * objects,
    int output_mode,
    int * num_extracted_objs,
    tag_t ** extracted_objs
)
```

tag_t	sketch_tag	Input	Sketch tag
int	count	Input	Number of objects to be extracted
tag_t *	objects	Input	Array of objects to be extracted
int	output_mode	Input	Output mode UF_SKET_EXTR_ORIG_TYPE UF_SKET_EXTR_MULTI_SPLINES UF_SKET_EXTR_SINGLE_SPLINE
int *	num_extracted_objs	Output	Number of extracted objects
tag_t **	extracted_objs	Output to UF_*free*	Array of extracted objects. Use UF_free to deallocate memory when done.

UF_SKET_add_objects ([view source](#))

Defined in: `uf_sket.h`

Overview

Adds geometric objects to the current sketch.

- The following restrictions applies:
- 1) UF_SKET_initialize_sketch must have been previously called.
 - 2) The construction coordinate system of the arcs you wish to add must be parallel to the sketch plane.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_add_objects
(
    tag_t sketch_tag,
    int count,
    tag_t * object
)
```

tag_t	sketch_tag	Input	Sketch tag.
int	count	Input	Number of objects to be added.
tag_t *	object	Input	Array of objects to be added to the sketch

UF_SKET_ask_active_sketch [\(view source\)](#)

Defined in: uf_sket.h

Overview

This routine returns the active sketch, or NULL_TAG if none active

Environment

Internal and External

History

This function was originally released in NX5.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_ask_active_sketch
(
    tag_t * sketch_tag
)
```

tag_t *	sketch_tag	Output	Active sketch tag, NULL_TAG if none
---------	------------	--------	-------------------------------------

UF_SKET_ask_con_is_inferred [\(view source\)](#)

Defined in: uf_sket.h

Overview

This function determines whether a given constraint is inferred or not.

Environment

Internal and External

History

This function was originally released in V18.0.

Required License(s)

gateway

```
int UF_SKET_ask_con_is_inferred
(
    tag_t con_tag,
    logical * inferred_con_fl
)
```

tag_t	con_tag	Input	Constraint object tag.
logical *	inferred_con_fl	Output	A flag that determines whether a given con is inferred or not.

UF_SKET_ask_constraint_class [\(view source\)](#)

Defined in: uf_sket.h

Overview
This function returns the constaint class of the given constraint tag.

Environment
Internal and External

History
This function was originally released in V17.0.

Required License(s)
gateway

```
int UF_SKET_ask_constraint_class
(
    tag_t con_tag,
    UF_SKET_con_class_t * con_class
)
```

tag_t	con_tag	Input	Constraint tag
UF_SKET_con_class_t *	con_class	Output	Constraint class: - UF_SKET_geo_cons - UF_SKET_dim_cons

UF_SKET_ask_constraint_type [\(view source\)](#)

Defined in: uf_sket.h

Overview
This function finds the constaint type

Environment

Internal and External

History

This function was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_constraint_type
(
    tag_t con_tag,
    UF_SKET_con_type_t * con_type
)
```

tag_t	con_tag	Input	Constraint tag
UF_SKET_con_type_t *	con_type	Output	The type of constraint

UF_SKET_ask_constraints_of_geometry (view source)

Defined in: uf_sket.h

Overview

This function finds all the constraints, with the specified class, associated with a geometry.

Environment

Internal and External

See Also

[UF_SKET_ask_geo_cons_of_geometry](#)

History

V17 change: This function is new for V17.0. Although the function's name is not new, the parameter list has changed.

Past callers of the UF_SKET_ask_constraints_of_geometry function should replace their calls by UF_SKET_ask_geo_cons_of_geometry or may use this function with additional parameter 'con_class' set to SKET_geo_cons to get the same result as before.

Required License(s)

gateway

```
int UF_SKET_ask_constraints_of_geometry
(
    tag_t sketch_tag,
    tag_t geom_tag,
    UF_SKET_con_class_t con_class,
    int * con_num,
    tag_t ** con_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
-------	------------	-------	------------

tag_t	geom_tag	Input	Tag of the geometry.
UF_SKET_con_class_t	con_class	Input	Constraint class: - UF_SKET_all_cons - UF_SKET_geo_cons - UF_SKET_dim_cons
int *	con_num	Output	Number of output constraints
tag_t **	con_tags	Output to UF_*free*	Array of constraints of the sketch. Use UF_free to deallocate memory when done.

UF_SKET_ask_constraints_of_sketch [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Output constraints, with the specified class, of a given sketch.

Environment

Internal and External

See Also

[UF_SKET_ask_geo_cons_of_sketch](#)

History

V17 change: This function is new for V17.0. Although the function's name is not new, the parameter list has changed.

Past callers of the `UF_SKET_ask_constraints_of_sketch` function should replace their calls by `UF_SKET_ask_geo_cons_of_sketch` or may use this function with additional parameter `'con_class'` set to `SKET_geo_cons` to get the same result as before.

Required License(s)

gateway

```
int UF_SKET_ask_constraints_of_sketch
(
    tag_t sketch_tag,
    UF_SKET_con_class_t con_class,
    int * num_cons,
    tag_t ** con_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
UF_SKET_con_class_t	con_class	Input	Constraint class: - UF_SKET_all_cons - UF_SKET_geo_cons - UF_SKET_dim_cons
int *	num_cons	Output	Number of output constraints

<code>tag_t **</code>	<code>con_tags</code>	Output to UF_*free*	Array of constraints of the sketch. Use UF_free to deallocate memory when done.
-----------------------	-----------------------	---------------------	--

UF_SKET_ask_dim_status [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Returns status of a sketch dimension as well as its expression.

Environment

Internal and External

Required License(s)

gateway

```
int UF_SKET_ask_dim_status
(
    tag_t dim_tag,
    tag_t * exp_tag,
    char exp_string [ 256 ] ,
    double * value,
    int * status
)
```

<code>tag_t</code>	<code>dim_tag</code>	Input	Dimension tag.
<code>tag_t *</code>	<code>exp_tag</code>	Output	Expression tag.
<code>char</code>	<code>exp_string [256]</code>	Output	Expression string.
<code>double *</code>	<code>value</code>	Output	Value of expression.
<code>int *</code>	<code>status</code>	Output	Status: 1 = Reference 2 = Constrained(for pre-v13 sketches) or Active(for v13+ sketches) 3 = Underconstrained(for pre-v13 sketches only)- some variables associated with the dimension are not constrained)

UF_SKET_ask_dimensions_of_sketch [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Output dimensions of a given sketch.

Environment

Internal and External

History

This function was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_dimensions_of_sketch
(
    tag_t sketch_tag,
    int * num_dims,
    tag_t ** dim_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
int *	num_dims	Output	Number of output dimensions
tag_t **	dim_tags	Output to UF_*free*	Array of dimensions of the sketch. Use UF_free to deallocate memory when done.

UF_SKET_ask_exps_of_sketch [\(view source\)](#)

Defined in: uf_sket.h

Overview

Returns the expressions for a sketch.

Environment

Internal and External

History

Original release was in V14.0.

Required License(s)

gateway

```
int UF_SKET_ask_exps_of_sketch
(
    tag_t sketch_tag,
    int * num_exps,
    tag_t ** expression_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
int *	num_exps	Output	Number of expressions in sketch
tag_t **	expression_tags	Output to UF_*free*	Array of expressions from the sketch. Use UF_free to deallocate memory when done.

UF_SKET_ask_face_sketches (view source)

Defined in: uf_sket.h

Overview

Returns all sketches associated with a face or datum plane.

Environment

Internal and External

Required License(s)

gateway

```
int UF_SKET_ask_face_sketches
(
    tag_t object,
    uf_list_p_t * object_list
)
```

tag_t	object	Input	Solid face / datum plane tag.
uf_list_p_t *	object_list	Output to UF_*free*	List of sketches or NULL if none found. This argument must be freed by calling UF_MODL_delete_list

UF_SKET_ask_feature_sketches (view source)

Defined in: uf_sket.h

Overview

Returns all sketches associated with a feature.

Environment

Internal and External

Required License(s)

gateway

```
int UF_SKET_ask_feature_sketches
(
    tag_t feature,
    uf_list_p_t * object_list
)
```

tag_t	feature	Input	Feature tag.
uf_list_p_t *	object_list	Output to UF_*free*	List of sketches or NULL if none found. This argument must be freed by calling UF_MODL_delete_list

UF_SKET_ask_geo_cons_of_geometry [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function finds all the geometric constraints associated with a geometry.

Environment

Internal and External

See Also

[UF_SKET_ask_constraints_of_geometry](#)

History

V17 change: This function was renamed from `UF_SKET_ask_constraints_of_geometry`, which was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_geo_cons_of_geometry
(
    tag_t sketch_tag,
    tag_t geom_tag,
    int * con_num,
    tag_t ** con_tags
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch tag.
<code>tag_t</code>	<code>geom_tag</code>	Input	Tag of the geometry.
<code>int *</code>	<code>con_num</code>	Output	Number of constraints.
<code>tag_t **</code>	<code>con_tags</code>	Output to UF_*free*	Array of constraints. Use UF_free to deallocate memory when done.

UF_SKET_ask_geo_cons_of_sketch [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Output geometric constraints of a given sketch.

Environment

Internal and External

See Also

[UF_SKET_ask_constraints_of_sketch](#)

History

V17 change: This function was renamed from `UF_SKET_ask_constraints_of_sketch`, which was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_geo_cons_of_sketch
(
    tag_t sketch_tag,
    int * num_cons,
    tag_t ** con_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
int *	num_cons	Output	Number of output constraints
tag_t **	con_tags	Output to UF_*free*	Array of constraints of the sketch. Use UF_free to deallocate memory when done.

UF_SKET_ask_geoms_of_sketch

([view source](#))

Defined in: `uf_sket.h`

Overview

Output geometries of a given sketch.

Environment

Internal and External

History

This function was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_geoms_of_sketch
(
    tag_t sketch_tag,
    int * num_geoms,
    tag_t ** geom_tags
)
```

tag_t	sketch_tag	Input	Sketch tag
int *	num_geoms	Output	Number of output geometries
tag_t **	geom_tags	Output to UF_*free*	Array of geometries of the sketch. Use UF_free to deallocate memory when done.

UF_SKET_ask_inferred_cons_of_sketch

([view source](#))

Defined in: `uf_sket.h`

Overview

This function returns all inferred constraints of the given sketch.

Environment

Internal and External

History

This function was originally released in V18.0.

Required License(s)

gateway

```
int UF_SKET_ask_inferred_cons_of_sketch
(
    tag_t sketch_tag,
    int * num_cons,
    tag_t ** con_tags
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch object tag.
<code>int *</code>	<code>num_cons</code>	Output	Number of inferred cons.
<code>tag_t **</code>	<code>con_tags</code>	Output to UF_*free*	Array of inferred cons. Use UF_free to deallocate the memory when done.

UF_SKET_ask_legacy_preferences [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Returns current sketch preference settings for sketches created prior to V13.0.

Environment

Internal and External

See Also

[UF_SKET_ask_preferences](#)
to query preferences for sketches created in V13.0 and beyond.
[UF_SKET_set_legacy_preferences](#)
[UF_SKET_set_preferences](#)

History

V15.0 change: This function was renamed from
UF_SKET_ask_preferences to UF_SKET_ask_legacy_preferences.

Required License(s)

gateway

```
int UF_SKET_ask_legacy_preferences
(
```

```

double * snap_angle,
double * cap_dist,
char pt_name [ 256 ] ,
int auto_flag [ 2 ] ,
int show_flag [ 3 ] ,
double * char_size,
int * dec_places,
int * ext_lines,
int * dim_label
)

```

double *	snap_angle	Output	Snap angle.
double *	cap_dist	Output	Capture distance.
char	pt_name [256]	Output	Point name variable.
int	auto_flag [2]	Output	Auto inferencing/constraint flag (1 = Off, 2 = On): [0] = Auto inferencing [1] = Auto constraint
int	show_flag [3]	Output	Show csys / datum / arrows flag (1 = Off, 2 = On): [0] = CSYS [1] = Datum [2] = Arrows
double *	char_size	Output	Character size.
int *	dec_places	Output	Decimal places.
int *	ext_lines	Output	Extension lines: 1 = Both 2 = Line1 3 = Line2 4 = None
int *	dim_label	Output	Dimension label: 1 = Value 2 = Expression 3 = Name

UF_SKET_ask_preferences [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Returns the current values for each of the sketch preference variables applicable for sketches created in V13.0 and beyond.

To determine if this function should be called instead of `UF_SKET_ask_legacy_preferences`, use `UF_OBJ_ask_type_and_subtype` and check for `UF_v13_sketch_subtype`.

Environment

Internal and External

See Also

[UF_SKET_ask_legacy_preferences](#)
[UF_SKET_set_preferences](#)

UF_SKET_set_legacy_preferences

History

V15.0 change: This function is new for V15.0. Although the function's name is not new, the parameter list has changed.

NX6 change: Sketches no longer have a 'character size' and 'decimal places' preference. Each individual dimension will have these preferences. The parameter list for this function will not change, but the values returned for 'character size' and 'decimal places' will always be the global annotation preferences.

Required License(s)

(solid_modeling or drafting)

int UF_SKET_ask_preferences

```
(
    tag_t sketch_tag,
    double * snap_angle,
    char name_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char vertex_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char line_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char arc_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char conic_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char spline_prefix [ UF_OBJ_NAME_BUFSIZE ],
    double * char_size,
    int * dec_places,
    int * dim_label
)
```

tag_t	sketch_tag	Input	Tag of input sketch; May be set to NULL_TAG if there is not a particular sketch which the user wants to query.
double *	snap_angle	Output	Snap angle preference
char	name_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for new sketch names
char	vertex_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for vertex names
char	line_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for line names
char	arc_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for arc names
char	conic_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for conic names
char	spline_prefix [UF_OBJ_NAME_BUFSIZE]	Output	Prefix for spline names
double *	char_size	Output	Character size is no longer a sketch preference. This parameter will always return the global annotation preferences- dimension character size

int *	dec_places	Output	Decimal places are no longer a sketch preference. This parameter will always return the global annotation preferences- dimension decimal places
int *	dim_label	Output	Dimension Label: 1 = value 2 = expression 3 = name

UF_SKET_ask_reference_status [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function returns the reference status - reference/active status of a sketch curve or a dimension

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_set_reference_status](#)

History

This function was originally released in V17.0.

Required License(s)

gateway

```
int UF_SKET_ask_reference_status
(
    tag_t skt_tag,
    tag_t member,
    UF_SKET_reference_status_t * status
)
```

tag_t	skt_tag	Input	Tag of the sketch to which the dim/curve belongs
tag_t	member	Input	Tag of the dim/curve whose state needs to be determined
UF_SKET_reference_status_t *	status	Output	UF_SKET_active or UF_SKET_reference

UF_SKET_ask_sket_freq_eid [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function returns a sketch feature eid given a sketch curve eid.

Environment

Internal and External

History

This function was originally released in V18.0.

Required License(s)

gateway

```
int UF_SKET_ask_sketch_freq_eid
(
    tag_t sket_eid,
    tag_t * sket_freq_eid
)
```

<code>tag_t</code>	<code>sket_eid</code>	Input	Sketch curve tag
<code>tag_t *</code>	<code>sket_freq_eid</code>	Output	Sketch feature tag

UF_SKET_ask_sketch_features [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Returns all features associated with a sketch.

Environment

Internal and External

Required License(s)

gateway

```
int UF_SKET_ask_sketch_features
(
    tag_t sketch_tag,
    uf_list_p_t * object_list
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch tag.
<code>uf_list_p_t *</code>	<code>object_list</code>	Output to UF_*free*	List of sketches or NULL if none found. This argument must be freed by calling UF_MODL_delete_list

UF_SKET_ask_sketch_info

([view source](#))

Defined in: `uf_sket.h`

Overview
Output general information about a given sketch.

Environment
Internal and External

History
This function was originally released in V16.0.

Required License(s)
gateway

```
int UF_SKET_ask_sketch_info
(
    tag_t sketch_tag,
    UF_SKET_info_t * sket_info
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch tag
<code>UF_SKET_info_t *</code>	<code>sket_info</code>	Output	Sketch information data structure. (See type <code>UF_SKET_info_t</code> in <code>uf_sket_types.h</code> for the structure members).

UF_SKET_ask_sketch_of_geom

([view source](#))

Defined in: `uf_sket.h`

Overview
Output sketch of a given geometric object

Environment
Internal and External

History
This function was originally released in NX 4.

Required License(s)
gateway

```
int UF_SKET_ask_sketch_of_geom
(
    tag_t geom_tag,
    tag_t * sketch_tag
)
```

<code>tag_t</code>	<code>geom_tag</code>	Input	Geometric object
--------------------	-----------------------	-------	------------------

<code>tag_t *</code>	<code>sketch_tag</code>	Output	Sketch tag of which object is member NULL_TAG: if none
----------------------	-------------------------	--------	---

UF_SKET_ask_sketch_status [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This routine returns the given sketch's status and the degrees of freedom needed to make the sketch full-constrained if it is currently under-constrained.

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

History

This function was originally released in V16.0.

Required License(s)

gateway

```
int UF_SKET_ask_sketch_status
(
    tag_t sketch_tag,
    UF_SKET_status_t * sket_status,
    int * dof_needed
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch tag
<code>UF_SKET_status_t *</code>	<code>sket_status</code>	Output	Sketch's current status. (See type <code>UF_SKET_status_t</code> in <code>uf_sket_types.h</code> for valid values).
<code>int *</code>	<code>dof_needed</code>	Output	Degrees of freedom needed to make the sketch fully-constrained. This value is meaningful only when sketch's status is <code>UF_SKET_under_constrained</code> .

UF_SKET_attach_to_face [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Attaches a non-feature sketch to a planar face or datum plane and returns the resulting sketch feature tag. For reattaching sketch features to a different

face or datum plane, please use UF_MODL_reattach_target_face.

Environment

Internal and External

See Also

[UF_MODL_reattach_target_face](#)

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_attach_to_face
(
    tag_t sketch_tag,
    tag_t face_tag,
    tag_t ref_tag,
    int ref_info [ 2 ] ,
    int plane_dir,
    tag_t * sketch_feature_tag
)
```

tag_t	sketch_tag	Input	Tag of the sketch to attach
tag_t	face_tag	Input	Face or datum plane to attach the sketch to
tag_t	ref_tag	Input	This defines a line which is the horizontal or vertical reference used for positioning the sketch on the face or datum plane. This object can be a linear edge, a datum axis, a planar face, or a datum plane. The direction of the line is specified in ref_info. A face or datum plane constructs the line by intersecting itself with face_tag.
int	ref_info [2]	Input	Data required to fully define the reference. ref_info[0] = reference orientation use one of the following values: UF_SKET_HORIZONTAL UF_SKET_VERTICAL ref_info[1] = direction along reference use one of the following values: UF_SKET_ALONG_CURVE (start to end) UF_SKET_OPPOSITE_CURVE (end to start)
int	plane_dir	Input	Which side of plane or face to attach sketch. use one of the following values: UF_SKET_WITH_NORMAL (outwards from parent body) UF_SKET_OPPOSITE_NORMAL (inwards from parent body)
tag_t *	sketch_feature_tag	Output	Sketch feature tag created

UF_SKET_create_dimension [\(view source\)](#)

Defined in: uf_sket.h

Overview

This routine creates sketch dimensions of the specified type.
Not all parameters are required based on the type of dimension being created. Refer to the descriptions below to determine

which parameters are used for each type of sketch dimension.

The following restrictions applies:

UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

[UF_SKET_update_sketch](#)

[UF_SKET_read_dimension](#)

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_create_dimension
(
    tag_t sketch_tag,
    UF_SKET_con_type_t dim_type,
    UF_SKET_dim_object_p_t dim_object1,
    UF_SKET_dim_object_p_t dim_object2,
    double dim_origin [ 3 ],
    tag_t * dim_tag
)
```

tag_t	sketch_tag	Input	Sketch tag
UF_SKET_con_type_t	dim_type	Input	Type of dimensions: UF_SKET_horizontal_dim UF_SKET_vertical_dim UF_SKET_parallel_dim UF_SKET_perpendicular_dim UF_SKET_angular_dim UF_SKET_radius_dim UF_SKET_diameter_dim
UF_SKET_dim_object_p_t	dim_object1	Input	Data of first object geometry tag.
UF_SKET_dim_object_p_t	dim_object2	Input	Data of second object geometry tag. Not used for UF_SKET_radius_dim or UF_SKET_diameter_dim.
double	dim_origin [3]	Input	Dimension origin (X, Y, Z) in model space
tag_t *	dim_tag	Output	Tag of created dimension

UF_SKET_create_dimensional_constraint [\(view source\)](#)

Defined in: [uf_sket.h](#)

Overview

This routine creates dimension constraint. This function differs from that of UF_SKET_create_dimension for it will output the constraint tag as opposed to the dimension tag and also the perimeter dimension creation is supported by this routine.

The following restrictions applies:

UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

[UF_SKET_update_sketch](#)

[UF_SKET_create_dimension](#)

[UF_SKET_create_dimensional_constraint](#)

History

This function was originally released in V17.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_create_dimensional_constraint
(
    tag_t sketch_tag,
    UF_SKET_con_type_t dim_type,
    int num_dim_obj,
    UF_SKET_dim_object_p_t dim_objs,
    double dim_origin [ 3 ],
    tag_t * con_tag
)
```

tag_t	sketch_tag	Input	Sketch tag
UF_SKET_con_type_t	dim_type	Input	Type of dimensions: UF_SKET_horizontal_dim UF_SKET_vertical_dim UF_SKET_parallel_dim UF_SKET_perpendicular_dim UF_SKET_angular_dim UF_SKET_radius_dim UF_SKET_diameter_dim UF_SKET_perimeter_dim
int	num_dim_obj	Input	Number of geometry tags in the geoms array
UF_SKET_dim_object_p_t	dim_objs	Input	Array of num_geom geometry tags
double	dim_origin [3]	Input	Dimension origin (X, Y, Z) in model space
tag_t *	con_tag	Output	Tag of created dimensional constraint

UF_SKET_create_geometric_constraint (view source)

Defined in: uf_sket.h

Overview

This routine creates sketch geometry constraints of the specified type. Not all parameters are required based on the type of constraint being created. Refer to the descriptions below to determine which parameters are used for each type of sketch constraints.

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

- UF_SKET_delete_constraints
- UF_SKET_initialize_sketch
- UF_SKET_read_geometric_constraint
- UF_SKET_update_sketch

History

V17 change: This function was renamed from UF_SKET_create_constraint, which was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_create_geometric_constraint
(
    tag_t sketch_tag,
    UF_SKET_con_type_t con_type,
    int num_con_geoms,
    UF_SKET_con_geom_t * con_geoms,
    tag_t * con_tag
)
```

tag_t	sketch_tag	Input	Sketch tag.
UF_SKET_con_type_t	con_type	Input	Constraint type. Valid types are: UF_SKET_fixed UF_SKET_horizontal UF_SKET_vertical UF_SKET_constant_length UF_SKET_constant_angle UF_SKET_uniform_scaled UF_SKET_non_uniform_scaled UF_SKET_parallel UF_SKET_perpendicular UF_SKET_collinear UF_SKET_equal_length UF_SKET_equal_radius UF_SKET_coincident UF_SKET_concentric UF_SKET_midpoint UF_SKET_slope UF_SKET_tangent UF_SKET_point_on_curve UF_SKET_point_on_string

int	num_con_geoms	Input	Number of constraint geometries
UF_SKET_con_geom_t *	con_geoms	Input	Array of constraint geometries (See type UF_SKET_con_geom_t in file uf_sket_types.h for the structure members).
tag_t *	con_tag	Output	The tag of the created geometric constraint.

UF_SKET_create_sketch (view source)

Defined in: uf_sket.h

Overview

Creates an empty sketch.

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_create_sketch
(
    char name [ UF_OBJ_NAME_BUFSIZE ] ,
    int option,
    double matrix [ 9 ] ,
    tag_t object [ 2 ] ,
    int reference [ 2 ] ,
    int plane_dir,
    tag_t * sketch_id
)
```

char	name [UF_OBJ_NAME_BUFSIZE]	Input	Sketch name It can be at most UF_OBJ_NAME_LEN bytes long
int	option	Input	Option. 1: Sketch on face/datum plane 2: Specify sketch CSYS
double	matrix [9]	Input	Sketch CSYS (for option = 2): [0-5]: X-AXIS and Y-AXIS of matrix [6-8]: Origin of CSYS
tag_t	object [2]	Input	Objects (for option = 1): [0]: Solid face/Datum plane object [1]: Reference object (edge, datum axis, solid face/datum

int	reference [2]	Input	Reference and direction (for option = 1): [0]: Reference edge 1: Horizontal 2: Vertical [1]: Direction 1: Start to end (from vertex1 to vertex2) -1: End to start (from vertex2 to vertex1)
int	plane_dir	Input	Datum plane direction: 1: Outwards from parent body. 2: Inward.
tag_t *	sketch_id	Output	Tag of teh sketch created

UF_SKET_delete_constraints [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This routine deletes constaints given their tags. The constraint tags could belong to different sketches.

Environment

Internal and External

See Also

[UF_SKET_create_geometric_constraint](#)
[UF_SKET_read_geometric_constraint](#)

History

This function was originally released in V16.0.

Required License(s)

(`solid_modeling` or `drafting`)

```
int UF_SKET_delete_constraints
(
    int num_cons,
    tag_t * con_tags
)
```

int	num_cons	Input	number of constraints to be deleted
tag_t *	con_tags	Input	Array of constraint tag.

UF_SKET_delete_dimensions [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This routine deletes dimensions given their tags. The dimension tags could belong to different sketches.

Environment

Internal and External

See Also

[UF_SKET_create_dimension](#)

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_delete_dimensions
(
    int num_dims,
    tag_t * dim_tags
)
```

int	num_dims	Input	Number of dimension tags to be deleted
tag_t *	dim_tags	Input	Array of dimension tags to be deleted

UF_SKET_delete_legacy_constraint [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Deletes the specified geometric constraint from an old sketch.

Environment

Internal and External

History

V16.0 change: This function was renamed from `UF_SKET_delete_constraint` to `UF_SKET_delete_legacy_constraint`.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_delete_legacy_constraint
(
    int type,
    tag_t obj_list [ 2 ] ,
    int assoc_var_list [ 2 ] ,
    int delete_all
)
```

int	type	Input	The type of geometric constraint, one of the following values: UF_SKET_DATUM UF_SKET_CONSTANT_OFFSETS UF_SKET_POINT_ON_CURVE UF_SKET_MIDPOINT_OF_CURVE UF_SKET_HORIZONTAL_LINES
-----	------	-------	--

			UF_SKET_VERTICAL_LINES UF_SKET_CONSTANT_ANGLES UF_SKET_CONSTANT_LENGTH_LINES UF_SKET_COLLINEAR UF_SKET_PARALLEL UF_SKET_PERPENDICULAR UF_SKET_EQUAL_LENGTH UF_SKET_EQUAL_RADIUS UF_SKET_TANGENT_CURVES
tag_t	obj_list [2]	Input	When delete_all flag = 0, the tag of the object from which the constraint is deleted.
int	assoc_var_list [2]	Input	When delete_all flag = 0, the variable index associated with the object if type = UF_SKET_DATUM UF_SKET_CONSTANT_OFFSET UF_SKET_POINT_ON_CURVE UF_SKET_MIDPOINT_OF_CURVE UF_SKET_CONSTANT_ANGLES UF_SKET_CONSTANT_LENGTH_LINES UF_SKET_TANGENT_CURVES UF_SKET_CONSTANT_OFFSET UF_SKET_TANGENT_CURVES NOT USED for type = UF_SKET_HORIZONTAL_LINES UF_SKET_VERTICAL_LINES UF_SKET_COLLINEAR UF_SKET_PARALLEL UF_SKET_PERPENDICULAR UF_SKET_EQUAL_LENGTH UF_SKET_EQUAL_RADIUS
int	delete_all	Input	Delete all flag, where 1 = yes (delete all constraints of specified type)

UF_SKET_initialize_sketch [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Initializes the sketch environment.

Environment

Internal and External

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_initialize_sketch
(
    char name [ UF_OBJ_NAME_BUFSIZE ] ,
    tag_t * object
)
```

char	name [UF_OBJ_NAME_BUFSIZE]	Input / Output	Input sketch name, returns updated sketch name. It should hold UF_OBJ_NAME_LEN bytes plus the trailing null.
------	-------------------------------------	----------------	--

<code>tag_t *</code>	<code>object</code>	Output	Sketch tag, NULL_TAG means non-existent sketch.
----------------------	---------------------	--------	---

UF_SKET_is_out_of_date [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function returns TRUE if the given sketch has any out of date references. This function is valid only for modeling sketches created in V13.0 and later versions of NX.

Environment

Internal and External

History

This function was originally released in V4.0.3.

Required License(s)

gateway

```
int UF_SKET_is_out_of_date
(
    tag_t sket_eid,
    logical * out_of_date
)
```

<code>tag_t</code>	<code>sket_eid</code>	Input	Sketch tag
<code>logical *</code>	<code>out_of_date</code>	Output	TRUE => The sketch has out-of-date references

UF_SKET_mirror_objects [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function mirrors the objects on a center line. New objects and "mirror" constraints between the original objects and the new objects will be created. Note that points on the center line and lines collinear to the center line will not be mirrored. Therefore, the output number of mirrored objects will not necessarily be the same as the input number of objects.

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_mirror_objects
(
    tag_t sketch_tag,
    tag_t center_line_tag,
    int num_objs,
    tag_t * obj_tags,
    int * num_new_objs,
    tag_t ** new_obj_tags,
    tag_t ** con_tags
)
```

tag_t	sketch_tag	Input	Sketch tag.
tag_t	center_line_tag	Input	Tag of the center line for mirror
int	num_objs	Input	Number of objects.
tag_t *	obj_tags	Input	Array of objects to be mirrored
int *	num_new_objs	Output	Number of mirrored objects or constraints created.
tag_t **	new_obj_tags	Output to UF_*free*	Array of mirrored objects. Use UF_free to deallocate memory when done.
tag_t **	con_tags	Output to UF_*free*	Array of constraints. Use UF_free to deallocate memory when done.

UF_SKET_read_dimension [\(view source\)](#)

Defined in: uf_sket.h

Overview

This function queries the dimension information.

Environment

Internal and External

See Also

[UF_SKET_create_dimension](#)

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```

int UF_SKET_read_dimension
(
    tag_t sketch_tag,
    tag_t dim_tag,
    UF_SKET_con_type_t * dim_type,
    UF_SKET_dim_object_p_t dim_object1,
    UF_SKET_dim_object_p_t dim_object2,
    double dim_origin [ 3 ],
    tag_t * dim_exp
)

```

tag_t	sketch_tag	Input	Sketch tag
tag_t	dim_tag	Input	Dimension tag
UF_SKET_con_type_t *	dim_type	Output	Type of dimensions: UF_SKET_horizontal_dim UF_SKET_vertical_dim UF_SKET_parallel_dim UF_SKET_perpendicular_dim UF_SKET_angular_dim UF_SKET_radius_dim UF_SKET_diameter_dim
UF_SKET_dim_object_p_t	dim_object1	Output	Data of first object geometry tag
UF_SKET_dim_object_p_t	dim_object2	Output	Data of second object geometry tag. Not used for UF_SKET_radius_dim or UF_SKET_diameter_dim
double	dim_origin [3]	Output	Dimension origin in model space
tag_t *	dim_exp	Output	Dimension expression tag (NULL_TAG for Reference dimensions)

UF_SKET_read_dimensional_constraint [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This routine reads dimension constraint given a constraint tag. This routine differs from that of `UF_SKET_read_dimension` as it takes in the constraint tag as opposed to the dimension tag.

The following restrictions applies:

`UF_SKET_initialize_sketch` must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_initialize_sketch](#)
[UF_SKET_update_sketch](#)
[UF_SKET_read_dimension](#)
[UF_SKET_read_dimensional_constraint](#)

History

This function was originally released in V17.0.

Required License(s)
(solid_modeling or drafting)

```
int UF_SKET_read_dimensional_constraint
(
    tag_t sketch_tag,
    tag_t con_tag,
    UF_SKET_con_type_t * dim_type,
    int * num_dim_obj,
    UF_SKET_dim_object_p_t * dim_objs,
    double dim_origin [ 3 ],
    tag_t * dim_tag,
    tag_t * exp_tag
)
```

tag_t	sketch_tag	Input	Sketch tag
tag_t	con_tag	Input	Tag of dimensional constraint to read.
UF_SKET_con_type_t *	dim_type	Output	Dimension type
int *	num_dim_obj	Output	Number of geometry tags in the geoms array
UF_SKET_dim_object_p_t *	dim_objs	Output to UF_*free*	Array of num_geom geometry tags Use UF_free to deallocate memory when done.
double	dim_origin [3]	Output	Dimension origin (X, Y, Z) in model space
tag_t *	dim_tag	Output	Tag of associated dimension object. Could be a NULL_TAG for cases such as perimeter dimensions.
tag_t *	exp_tag	Output	Tag of associated expression object

UF_SKET_read_geometric_constraint [\(view source\)](#)

Defined in: uf_sket.h

Overview

This function queries the constraint information

Environment

Internal and External

See Also

[UF_SKET_create_geometric_constraint](#)

History

V17 change: This function was renamed from UF_SKET_read_constraint, which was originally released in V16.0.

Required License(s)
(solid_modeling or drafting)

```
int UF_SKET_read_geometric_constraint
(
    tag_t sketch_tag,
    tag_t con_tag,
    UF_SKET_con_type_t * con_type,
    int * geom_count,
    UF_SKET_con_geom_p_t * con_geoms
)
```

tag_t	sketch_tag	Input	Sketch tag.
tag_t	con_tag	Input	Constraint tag.
UF_SKET_con_type_t *	con_type	Output	Constraint type.
int *	geom_count	Output	Number of geometries.
UF_SKET_con_geom_p_t *	con_geoms	Output to UF_*free*	Array of constraint geometries. Use UF_free to deallocate memory when done.

UF_SKET_set_legacy_preferences [\(view source\)](#)

Defined in: uf_sket.h

Overview
Modify the current sketch preference settings for sketches created prior to V13.0.

Environment
Internal and External

See Also
[UF_SKET_ask_preferences](#)
[UF_SKET_ask_legacy_preferences](#)
[UF_SKET_set_preferences](#)

History
V15.0 change: This function was renamed from UF_SKET_set_preferences to UF_SKET_set_legacy_preferences.

Required License(s)
or drafting)>

```
int UF_SKET_set_legacy_preferences
(
    const int values [ 9 ] ,
    double snap_angle,
    double cap_dist,
    const char * pt_name,
    const int auto_flag [ 2 ] ,
```

```

const int show_flag [ 3 ],
double char_size,
int dec_places,
int ext_lines,
int dim_label
)

```

const int	values [9]	Input	Array of flags to indicate which settings should be modified (0 = leave at current value, 1 = change to new value): [0] = Snap angle [1] = Capture distance [2] = Point name variable [3] = Auto inferencing/constraint flag [4] = Show csys/datum/arrows flag [5] = Character size [6] = Decimal places [7] = Extension lines [8] = Dimension label
double	snap_angle	Input	Snap angle.
double	cap_dist	Input	Capture distance.
const char *	pt_name	Input	Point name variable.
const int	auto_flag [2]	Input	Auto inferencing/constraint flag (1= Off; 2= On): [0] = Auto inferencing [1] = Auto constraint
const int	show_flag [3]	Input	Show csys / datum / arrows flag (1= Off; 2= On) [0] = CSYS [1] = Datum [2] = Arrows
double	char_size	Input	Character size.
int	dec_places	Input	Decimal places.
int	ext_lines	Input	Extension lines: 1 = Both 2 = Line1 3 = Line2 4 = None
int	dim_label	Input	Dimension label: 1 = Value 2 = Expression 3 = Name

UF_SKET_set_preferences [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

Sets the sketch preferences for sketches created in V13 and beyond.

To determine if this function should be used instead of

UF_SKET_set_legacy_preferences, use UF_OBJ_ask_type_and_subtype

and check for subtype UF_v13_sketch_subtype.

Environment

Internal and External.

See Also

[UF_SKET_ask_preferences](#)
to set preferences for sketches created in V13.0 and beyond.
[UF_SKET_ask_legacy_preferences](#)
[UF_SKET_set_legacy_preferences](#)

History

This function is new for V15.0. Although the function name is not new, the parameter list has changed.

NX6 change: Sketches no longer have a 'character size' and 'decimal places' preference. Each individual dimension will have these preferences. The parameter list for this function has not changed. Calling this function will result in the 'character size' and 'decimal places' being set on each individual dimension in the sketch.

Required License(s)

gateway

```
int UF_SKET_set_preferences
(
    tag_t sketch_tag,
    int values [ 10 ],
    double snap_angle,
    char name_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char vertex_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char line_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char arc_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char conic_prefix [ UF_OBJ_NAME_BUFSIZE ],
    char spline_prefix [ UF_OBJ_NAME_BUFSIZE ],
    double char_size,
    int dec_places,
    int dim_label
)
```

tag_t	sketch_tag	Input	Sketch tag. May be NULL_TAG if there is no existing sketch to which the preferences are to be associated. The preferences will then be applied to the sketches created thereafter, with the exception of character size and decimal places. These two will always come from annotation preferences.
int	values [10]	Input	Array flag to indicate what action to perform for which sketch preference variable. 0: Do nothing 1: Action [0]: Snap angle [1]: Sketch name prefix [2]: Vertex name prefix [3]: Line name prefix [4]: Arc name prefix [5]: Conic name prefix [6]: Spline name prefix

[7]: Character size
 [8]: Decimal places
 [9]: Dimension label

double	snap_angle	Input	Snap angle
char	name_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Sketch name prefix
char	vertex_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Vertex name prefix
char	line_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Line lname prefix
char	arc_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Arc name prefix
char	conic_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Conic name prefix
char	spline_prefix [UF_OBJ_NAME_BUFSIZE]	Input	Spline name prefix
double	char_size	Input	Character size is no longer a sketch preference. The character size will be set for each individual dimension in the sketch. If sketch_tag==NULL_TAG nothing will be done.
int	dec_places	Input	Decimal places are no longer a sketch preference. The decimal places will be set for each individual dimension in the sketch. If sketch_tag==NULL_TAG nothing will be done.
int	dim_label	Input	Dimension label 1 : Value 2 : Expression 3 : Name

UF_SKET_set_reference_status [\(view source\)](#)

Defined in: `uf_sket.h`

Overview

This function sets the dimension/sketch curve to reference/active.

The following restrictions applies:

UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

[UF_SKET_ask_reference_status](#)

History

This function was originally released in V17.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_set_reference_status
(
    tag_t skt_tag,
    tag_t member,
    UF_SKET_reference_status_t status
)
```

tag_t	skt_tag	Input	Tag of the sketch to which the dim/curve belongs
tag_t	member	Input	Array of the dim/curve tags whose state has to be set
UF_SKET_reference_status_t	status	Input	UF_SKET_active or UF_SKET_reference

UF_SKET_terminate_sketch (view source)

Defined in: uf_sket.h

Overview

Terminate the current sketch.

The following restrictions applies:
UF_SKET_initialize_sketch must have been previously called.

Note: When in Drafting it is not possible to terminate the sketch.

Environment

Internal and External

See Also

- UF_SKET_initialize_sketch
- UF_SKET_update_sketch

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_terminate_sketch
(
    void
)
```

UF_SKET_update_sketch (view source)

Defined in: uf_sket.h

Overview

This function updates the current sketch. If you had call to UF_SKET_create_geometric_constraint, UF_SKET_create_dimension or UF_SKET_mirror_objects to create constraints or dimensions in a sketch, it is highly recommended that this routine should be called before calling UF_SKET_terminate_sketch.

The following restrictions applies:

UF_SKET_initialize_sketch must have been previously called.

Environment

Internal and External

See Also

UF_SKET_initialize_sketch
UF_SKET_create_dimension
UF_SKET_create_geometric_constraint
UF_SKET_mirror_objects
UF_SKET_terminate_sketch

History

This function was originally released in V16.0.

Required License(s)

(solid_modeling or drafting)

```
int UF_SKET_update_sketch  
(  
    tag_t sketch_tag  
)
```

<code>tag_t</code>	<code>sketch_tag</code>	Input	Sketch object tag.
--------------------	-------------------------	-------	--------------------