

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

Defined in: `uf_point.h`

Overview

Ask the point tag from an associative point feature.

Environment

Internal and External

Required License(s)

gateway

```
int UF_POINT_ask_point_output
(
    tag_t point_feature_id,
    tag_t * point_id
)
```

<code>tag_t</code>	<code>point_feature_id</code>	Input	Object id of associative point feature.
<code>tag_t *</code>	<code>point_id</code>	Output	Object id of the point created by the feature.

UF_POINT_create_3_scalars [\(view source\)](#)

Defined in: `uf_point.h`

Overview

Creates an associative point with three scalars.

Return

error code

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_3_scalars
(
    const tag_t xyz [ 3 ],
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	<code>xyz [3]</code>	Input	Array of object ids of x,y, and z scalars
<code>tag_p_t</code>	<code>point_feature_id</code>	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point along curve using a curve with base point and t (scalar offset along curve). This point is derived by finding the closest point on the given curve to the given base point and then offsetting this point along the given curve by the arc length defined by t at absolute distance or relative percent.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_along_curve
(
    const tag_t curve,
    const tag_t base_point,
    const tag_t t,
    const UF_SO_point_along_curve_option_t option,
    const logical flip,
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	curve	Input	object id of curve or edge
<code>const tag_t</code>	base_point	Input	object id of base point
<code>const tag_t</code>	t	Input	object id of scalar offset along curve
<code>const UF_SO_point_along_curve_option_t</code>	option	Input	Absolute/Relative offset option. Can be one of the following enumerated constants: UF_SO_point_along_curve_distance UF_SO_point_along_curve_percent
<code>const logical</code>	flip	Input	If flip is set to TRUE, then $t = -t$
<code>tag_p_t</code>	point_feature_id	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point at the center of a conic.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_at_conic_center
(
    const tag_t conic,
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	<code>conic</code>	Input	object id of conic
<code>tag_p_t</code>	<code>point_feature_id</code>	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point at intersection of two curves.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_at_intersection_of_two_curves
(
    const tag_t curve1,
    const tag_t curve2,
    const tag_t help_point1,
    const tag_t help_point2,
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	<code>curve1</code>	Input	object id of first curve
<code>const tag_t</code>	<code>curve2</code>	Input	object id of second curve
<code>const tag_t</code>	<code>help_point1</code>	Input	object id of help point on first curve
<code>const tag_t</code>	<code>help_point2</code>	Input	object id of help point on second curve

<code>tag_p_t</code>	<code>point_feature_id</code>	Output	object id of point feature
----------------------	-------------------------------	--------	----------------------------

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

Defined in: `uf_point.h`

Overview

Creates an associative point on arc at an angle.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_on_arc_angle
(
    const tag_t arc,
    const tag_t angle,
    const tag_t xform,
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	<code>arc</code>	Input	object id of arc
<code>const tag_t</code>	<code>angle</code>	Input	object id of angle in radians
<code>const tag_t</code>	<code>xform</code>	Input	object id of optional transform
<code>tag_p_t</code>	<code>point_feature_id</code>	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point on curve at scalar value t.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_on_curve
(
    const tag_t curve,
    const tag_t t,
    tag_p_t point_feature_id
)
```

const tag_t	curve	Input	object id of curve
const tag_t	t	Input	object id of scalar
tag_p_t	point_feature_id	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point on a surface at uv scalars for the surface.

Note the u,v parameters are normalized from 0 to 1, so if you have read surface parameters with UF_MODL_ask_face_parms, or UF_MODL_ask_face_uv_minmax, you will have to normalize the parameters to the range of 0 to 1 prior to creating an associative point on the surface.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_on_surface
(
    const tag_t surface,
    const tag_t u,
    const tag_t v,
    tag_p_t point_feature_id
)
```

const tag_t	surface	Input	object id of surface
const tag_t	u	Input	object id of u scalar parameter
const tag_t	v	Input	object id of v scalar parameter
tag_p_t	point_feature_id	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point with surface and curve intersection

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_surface_curve_intersection
(
    const tag_t surface,
    const tag_t curve,
    const tag_t help_point1,
    const tag_t help_point2,
    tag_p_t point_feature_id
)
```

<code>const tag_t</code>	<code>surface</code>	Input	object id of surface
<code>const tag_t</code>	<code>curve</code>	Input	object id of curve or edge
<code>const tag_t</code>	<code>help_point1</code>	Input	object id of help point on surface
<code>const tag_t</code>	<code>help_point2</code>	Input	object id of help point on curve
<code>tag_p_t</code>	<code>point_feature_id</code>	Output	object id of point feature

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

Defined in: `uf_point.h`

Overview

Creates an associative point with a offset to the point.

Environment

Internal and External

See Also

[UF_POINT_ask_point_output](#)

Required License(s)

gateway

```
int UF_POINT_create_with_offset
(
```

```
const tag_t base_point,  
const tag_t offset,  
tag_p_t point_feature_id  
)
```

const tag_t	base_point	Input	object id of Base point to offset
const tag_t	offset	Input	object id of offset
tag_p_t	point_feature_id	Output	object id of point feature