UF_MODL_cliff_blend_f_t (view source)

Defined in: uf_modl_blends.h

Overview

Registers a routine for creating a cliff blend. This allows you to register a routine into the Open API environment for the cliff blend.

To register a routine named cliff_routine(), the call is:
UF_MODL_register_cliff_blend(cliff_routine)

where cliff_routine is defined as:
int cliff_routine(tag_t obj_id, int cliff)

Environment

Internal and External

```
int UF_MODL_cliff_blend_f_t
(
    tag_t,
    tag_t*
)
```

tag_t	Input
tag_t*	Input

UF_MODL_rpo_f_p_t (view source)

Defined in: uf_modl_types.h

Overview

Typedef for a rpo routine function pointer.

```
int UF_MODL_rpo_f_p_t
(
    tag_t
)
```

tag_t	Input

UF_MODL_udf_mapping_f_p_t (view source)

Defined in: uf_modl_types.h

Overview

Typedef for a UDF mapping routine function pointer.

```
int UF_MODL_udf_mapping_f_p_t
(
    tag_t,
    UF_MODL_udf_ref_data_p_t
)
```

tag_t
UF_MODL_udf_ref_data_p_t

UF_MODL_var_blend_f_t (view source)

Defined in: uf_modl_blends.h

Overview

Registers a routine for creating a variable radius blend (VRB). This allows you to register a routine into the Open API environment so that you can specify the points necessary for the VRB.

Subsequently, every time a blend is created, this registered routine is called. Note that this routine works in conjunction with UF_MODL_create_blend. The general procedure for using this function is to:

1. Register the vrb_routine with a call to UF_MODL_register_var_blend.
2. Create the variable blend with a call to UF_MODL_create_blend. Your registered routine is called once for each edge on the list that was inputted to UF_MODL_create_blend.
3. Unregister your vrb_routine with a call to UF_MODL_unregister_var_blend.

To register a routine named vrb_routine(), the call is:
UF_MODL_register_var_blend(vrb_routine)

where vrb_routine is defined as:

```
int vrb_routine(tag_t obj_id, double points[100][3],
char radii[100][256],
int smooth_overflow, int cliff_overflow,
int notch_overflow, int number_pts)
```

The integer values for smooth_overflow, cliff_overflow, and notch_overflow control the overflow during blending. The following string defined constants should be used.

UF_MODL_BLEND_NO_OVERFLOW - allows overflow control.
UF_MODL_BLEND_SMOOTH_OVERFLOW - does not allow smooth overflow control.
UF_MODL_BLEND_CLIFF_OVERFLOW - does not allow cliff overflow control.
UF_MODL_BLEND_NOTCH_OVERFLOW - does not allow notch overflow control.

vrb_tol allows you to specify a variable radius blend tolerance and it should be positive and bigger than 10E-8mm.

Environment

Internal and External

See Also

Refer to example

```
int UF_MODL_var_blend_f_t
(
    tag_t a,
    double b [ 100 ] [ 3 ] ,
    char c [ 100 ] [ 256 ] ,
    int smooth_overflow,
    int cliff_overflow,
    int notch_overflow,
    double vrb_tol,
    int * d
)
```

tag_t	**a**	Input
double	**b [100] [3]**	Input
char	**c [100] [256]**	Input
int	**smooth_overflow**	Input
int	**cliff_overflow**	Input
int	**notch_overflow**	Input
double	**vrb_tol**	Input
int *	**d**	Input