UF_MODL_add_rules_to_face_container (view source)

Defined in: uf_sc.h

Overview
Add the rules to an existing face container. If any rules are already in the face container, they are ignored.

Environment
Internal and External

See Also
UF_MODL_create_smart_face_container ,
UF_MODL_ask_rules_to_face_container ,
UF_MODL_remove_rules_from_face_container
Example: ufd_modl_face_containers.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_add_rules_to_face_container
(
    int n_rules,
    int * rule_types,
    UF_SC_input_data_t * rules,
    tag_t face_container_tag
)
```

int	**n_rules**	Input	The number of face container rules to be removed.
int *	**rule_types**	Input	The types of face container rules to be removed. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t *	**rules**	Input	The face container rules to be removed. The face container rule data are defined in uf_sc_types.h.
tag_t	**face_container_tag**	Input	The tag of the face container.

UF_MODL_add_rules_to_section (view source)

Defined in: uf_sc.h

Overview
Add section rules to an existing section. If any rules are already in the section, they are ignored.

Environment
Internal and External

See Also

UF_MODL_create_section ,
UF_MODL_ask_section ,
UF_MODL_remove_rules_from_section ,
UF_MODL_set_start_and_direction_to_section
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

int UF_MODL_add_rules_to_section
(
int n_rules,
UF_SC_section_data_t * rules,
tag_t section_tag
)

int	**n_rules**	Input	The number of section builder rules.
UF_SC_section_data_t *	**rules**	Input	An array of section builder rules. The section builder rule data are defined in uf_sc_types.h.
tag_t	**section_tag**	Input	The tag of the section builder.

UF_MODL_add_rules_to_wireframe_container (view source)

Defined in: uf_sc.h

Overview
Add the rules to an existing wireframe container. If any rules are already in the wireframe container, they are ignored.

Environment
Internal and External

See Also
UF_MODL_create_smart_wireframe_container ,
UF_MODL_ask_rules_to_wireframe_container ,
UF_MODL_remove_rules_from_wireframe_container
Example: ufd_modl_wireframe_containers.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

int UF_MODL_add_rules_to_wireframe_container
(
int n_rules,
int * rule_types,

```
        UF_SC_input_data_t * rules,
        tag_t wireframe_container_tag
    )
```

int	**n_rules**	Input	The number of wireframe container rules to be removed.
int *	**rule_types**	Input	The types of wireframe container rules to be removed. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t *	**rules**	Input	The wireframe container rules to be removed. The wireframe container rule data are defined in uf_sc_types.h.
tag_t	**wireframe_container_tag**	Input	The tag of the wireframe container.

UF_MODL_ask_container (view source)

Defined in: uf_sc.h

Overview
Returns the rule information and entities inside a container. For each rule, this function returns the type of rule and container rule data.

Environment
Internal and External

See Also
UF_MODL_ask_smart_face_container ,
UF_MODL_ask_smart_wireframe_container

History
This function was originally released in NX401.

Required License(s)
gateway

```
    int UF_MODL_ask_container
    (
        tag_t container_tag,
        int * n_rules,
        int * * rule_types,
        UF_SC_input_data_p_t * rules,
        int * n_entities,
        tag_t * * entities
    )
```

tag_t	**container_tag**	Input	The input container
int *	**n_rules**	Output	The number of rules in the container
int * *	**rule_types**	Output to UF_*free*	An integer array will be allocated. Each element in the array is the type of rule. The valid types are defined

			in uf_sc_types.h. The allocated memory needs to be freed.
UF_SC_input_data_p_t *	**rules**	Output to UF_*free*	An array of container rules will be allocated. The container rule data are defined in uf_sc_types.h. The allocated memory needs to be freed.
int *	**n_entities**	Output	The number of resulting entities.
tag_t * *	**entities**	Output to UF_*free*	An array of resulting entities will allocated. The entities will be either faces or wireframes or both. The allocated memory needs to be freed.

UF_MODL_ask_input_curves_from_section (view source)

Defined in: uf_sc.h

Overview

Returns the underlying curve and corresponding start and end connector for each of the output curves of a section, arranged in the same way as they are used in the loops of the section.

Environment

Internal and External

History

This function was originally released in NX4.

Required License(s)

gateway

```
int UF_MODL_ask_input_curves_from_section
(
    tag_t section,
    int * n_loops,
    int * * n_crv_each_loops,
    UF_SC_section_output_data_p_t * * input_curves
)
```

tag_t	**section**	Input	The input section
int *	**n_loops**	Output	The number of loops in the section. This argument cannot be NULL
int * *	**n_crv_each_loops**	Output to UF_*free*	An integer array will be allocated, the size of which is the number of loops in the section. Each element in the array is the number of curves in each loop. This argument cannot be NULL. The allocated memory needs to be freed.

UF_SC_section_output_data_p_t * *	**input_curves**	Output to UF_*free*	An array of an array of UF_SC_section_output_data_t will be allocated, where the ith row represents the ith loop, and the jth column in the ith row reprsents the input data for the jth curve in the ith loop. This argument can be NULL. The start and end connectors and the corresponding points in (input_curves)[i][j] will be wrt the corresponding loop direction. The allocated memory needs to be freed.

UF_MODL_ask_output_curves_from_section (view source)

Defined in: uf_sc.h

Overview

Return the number of curves, the curves' data structures, and the curves' positions in the input section.

Environment

Internal and External

See Also

UF_MODL_create_section ,
UF_MODL_ask_section ,
UF_MODL_free_section_data
UF_CURVE_free_curve_struct

History

This function was originally released in NX3.

Required License(s)

gateway

```
int UF_MODL_ask_output_curves_from_section
(
    tag_t section,
    int * n_curves,
    int * * indices,
    UF_CURVE_struct_p_t * * curves
)
```

tag_t	**section**	Input	The input section
int *	**n_curves**	Output	The number of curves in the section
int * *	**indices**	Input / Output to UF_*free*	If NULL is passed in, then nothing is returned. If a pointer is passed in, then an array of integers will be allocated. Each integer represents a position, starting from 0, of the output curve in the section. Do not change these integers. The array

			needed to be free.
UF_CURVE_struct_p_t * *	**curves**	Input / Output to UF_*free*	If NULL is passed in, then nothing is returned. If a pointer is passed in, then an array of UF_CURVE_struct_p_t will be allocated. Each pointer represents a output curve in the section. Do not change these pointers. Free the pointers using UF_CURVE_free_curve_struct. The array needed to be free.

UF_MODL_ask_section (view source)

Defined in: uf_sc.h

Overview

Return the section rules, the starting object, and the direction flag of a section container.

If you want to change the features that have section builders,
you should not delete the existing section builders and create
new ones. You should change the section builders using the
functions UF_MODL_add_rules_to_section and/or
UF_MODL_remove_rules_from_section and/or
UF_MODL_set_start_and_direction_to_section.

Environment

Internal and External

See Also

UF_MODL_create_section ,
UF_MODL_add_rules_to_section ,
UF_MODL_remove_rules_from_section ,
UF_MODL_set_start_and_direction_to_section
Example: ufd_modl_sections.c

History

This function was originally released in NX2.

Required License(s)

gateway

```
int UF_MODL_ask_section
(
    tag_t section_tag,
    int * n_rules,
    UF_SC_section_data_t * * rules,
    tag_t * starting_object,
    double starting_point [ 3 ] ,
    double direction [ 3 ]
)
```

tag_t	**section_tag**	Input	The tag of the section builder.
int *	**n_rules**	Output	The number of section builder rules.

UF_SC_section_data_t * *	**rules**	Output	An array of section builder rules. The section builder rule data are defined in uf_sc_types.h.
tag_t *	**starting_object**	Output	The object, curve or edge, specifying the starting of the section.
double	**starting_point [3]**	Input	The origin of the vector defined the direction of the section. This point has to be on the starting object.
double	**direction [3]**	Input	The direction of the vector defined the direction of the section.

UF_MODL_ask_smart_container_subtype (view source)

Defined in: uf_sc.h

Overview
Returns the subtype for an object of type UF_smart_container_type.
The subtypes are defined in uf_object_types.h

Environment
Internal and External

History
This function was originally released in NX4.

Required License(s)
solid_modeling

```
int UF_MODL_ask_smart_container_subtype
(
    tag_t smart_container_tag,
    int * smart_container_subtype
)
```

tag_t	**smart_container_tag**	Input	The tag of the smart container
int *	**smart_container_subtype**	Output	The subtype of smart container

---

UF_MODL_ask_smart_face_container (view source)

Defined in: uf_sc.h

Overview
Return the rules and the resulting faces of a face container.

If you want to change the features that have face containers,
you should not delete the existing face containers and create
new ones. You should change the face containers using the

functions UF_MODL_add_rules_to_face_container and/or UF_MODL_remove_rules_from_face_container.

Environment

Internal and External

See Also

UF_MODL_create_smart_face_container ,
UF_MODL_add_rules_to_face_container ,
UF_MODL_remove_rules_from_face_container
Example: ufd_modl_face_containers.c

History

This function was originally released in NX2.

Required License(s)

gateway

```
int UF_MODL_ask_smart_face_container
(
    tag_t face_container_tag,
    int * n_rules,
    int * * rule_types,
    UF_SC_input_data_t * * rules,
    int * n_face_eids,
    tag_t * * face_eids
)
```

tag_t	**face_container_tag**	Input	The tag of the face container.
int *	**n_rules**	Output	The number of face container rules.
int * *	**rule_types**	Output to UF_*free*	An array of the types for the face container rules. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t * *	**rules**	Output to UF_*free*	An array of face container rules. The face container rule data are defined in uf_sc_types.h.
int *	**n_face_eids**	Output	The number of resulting faces
tag_t * *	**face_eids**	Output to UF_*free*	An array of resulting faces

UF_MODL_ask_smart_wireframe_container (view source)

Defined in: uf_sc.h

Overview

Return the rules and the resulting wireframe, curves or edges, of a wireframe container.

If you want to change the features that have wireframe containers, you should not delete the existing wireframe containers and create new ones. You should change the wireframe containers using the functions UF_MODL_add_rules_to_wireframe_container and/or

UF_MODL_remove_rules_from_wireframe_container.

Environment
Internal and External

See Also
UF_MODL_create_smart_wireframe_container ,
UF_MODL_add_rules_to_wireframe_container ,
UF_MODL_remove_rules_from_wireframe_container
Example: ufd_modl_wireframe_containers.c

History
This function was originally released in NX2.

Required License(s)
gateway

```
int UF_MODL_ask_smart_wireframe_container
(
    tag_t wireframe_container_tag,
    int * n_rules,
    int * * rule_types,
    UF_SC_input_data_t * * rules,
    int * n_wireframe_eids,
    tag_t * * wireframe_eids
)
```

tag_t	**wireframe_container_tag**	Input	The tag of the wireframe container.
int *	**n_rules**	Output	The number of wireframe container rules.
int * *	**rule_types**	Output to UF_*free*	An array of the types for the wireframe container rules. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t * *	**rules**	Output to UF_*free*	An array of wireframe container rules. The wireframe container rule data are defined in uf_sc_types.h.
int *	**n_wireframe_eids**	Output	The number of resulting wireframes
tag_t * *	**wireframe_eids**	Output to UF_*free*	An array of resulting wireframes

UF_MODL_create_section (view source)

Defined in: uf_sc.h

Overview

Create a section builder from the input section rules. Return the error code.

If you want to change the features that have section builders,
you should not delete the existing section builders and create
new ones. You should change the section builders using the
functions UF_MODL_add_rules_to_section and/or
UF_MODL_remove_rules_from_section and/or
UF_MODL_set_start_and_direction_to_section.

Environment
Internal and External

See Also
UF_MODL_ask_section ,
UF_MODL_add_rules_to_section ,
UF_MODL_remove_rules_from_section ,
UF_MODL_set_start_and_direction_to_section
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_create_section
(
    tag_t object_in_part,
    int n_rules,
    UF_SC_section_data_t * rules,
    tag_t starting_object,
    double starting_point [ 3 ] ,
    double direction [ 3 ] ,
    logical allow_multiple_loops,
    tag_t * section_tag
)
```

tag_t	**object_in_part**	Input	Object in the part in which the section builder object is to be created.
int	**n_rules**	Input	The number of section rules.
UF_SC_section_data_t *	**rules**	Input	An array of section rules. The section rule data are defined in uf_sc_types.h.
tag_t	**starting_object**	Input	Optional. If this option is not used, then starting_object will be Zero. Otherwise, the object, curve or edge, specifying the starting of the section. If the section is a loop, then the loop is started by starting_object and the direction is followed the input direction. If the section is not a loop, then the section can be started by other object, but the direction of the section is defined by the starting_object and direction.
double	**starting_point [3]**	Input	If starting_object is used, the origin of the vector defined the direction of the section.

			This point has to be on the starting object.
double	**direction [3]**	Input	If starting_object is used, the direction of the vector defined the direction of the section.
logical	**allow_multiple_loops**	Input	If TRUE, the section may contain multiple loops.
tag_t *	**section_tag**	Output	The tag of the section builder

UF_MODL_create_smart_face_container (view source)

Defined in: uf_sc.h

Overview
Create a face container from the input rules. Return the error code.

If you want to change the features that have face containers,
you should not delete the existing face containers and create
new ones. You should change the face containers using the
functions UF_MODL_add_rules_to_face_container and/or
UF_MODL_remove_rules_from_face_container.

Environment
Internal and External

See Also
UF_MODL_ask_smart_face_container ,
UF_MODL_add_rules_to_face_container ,
UF_MODL_remove_rules_from_face_container
Example: ufd_modl_face_containers.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_create_smart_face_container
(
    tag_t object_in_part,
    int n_rules,
    int * rule_types,
    UF_SC_input_data_t * rules,
    tag_t * face_container_tag
)
```

tag_t	**object_in_part**	Input	Object in the part in which the face container object is to be created.
int	**n_rules**	Input	The number of face container rules.
int *	**rule_types**	Input	An array of the types for the face container rules. The valid types are defined in uf_sc_types.h.

UF_SC_input_data_t *	**rules**	Input	An array of face container rules. The face container rule data are defined in uf_sc_types.h.
tag_t *	**face_container_tag**	Output	The tag of the face container

UF_MODL_create_smart_wireframe_container (view source)

Defined in: uf_sc.h

Overview

Create a wireframe container from the input rules. Return the error code.

If you want to change the features that have wireframe containers, you should not delete the existing wireframe containers and create new ones. You should change the wireframe containers using the functions UF_MODL_add_rules_to_wireframe_container and/or UF_MODL_remove_rules_from_wireframe_container.

Environment

Internal and External

See Also

UF_MODL_ask_smart_wireframe_container ,
UF_MODL_add_rules_to_wireframe_container ,
UF_MODL_remove_rules_from_wireframe_container
Example: ufd_modl_wireframe_containers.c

History

This function was originally released in NX2.

Required License(s)

solid_modeling

```
int UF_MODL_create_smart_wireframe_container
(
    tag_t object_in_part,
    int n_rules,
    int * rule_types,
    UF_SC_input_data_t * rules,
    tag_t * wireframe_container_tag
)
```

tag_t	**object_in_part**	Input	Object in the part in which the wireframe container object is to be created.
int	**n_rules**	Input	The number of wireframe container rules.
int *	**rule_types**	Input	An array of the types for the wireframe container rules. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t *	**rules**	Input	An array of wireframe container rules. The wireframe container rule data are defined in uf_sc_types.h.

tag_t *	**wireframe_container_tag**	Output	The tag of the wireframe container

UF_MODL_free_sc_input_data (view source)

Defined in: uf_sc.h

Overview
Free the UF_MODL_free_sc_input_data.

Environment
Internal and External

See Also
UF_MODL_create_smart_face_container ,
UF_MODL_ask_smart_face_container ,
UF_MODL_create_smart_wireframe_container ,
UF_MODL_ask_smart_wireframe_container ,
UF_MODL_create_section ,
UF_MODL_ask_section
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_free_sc_input_data
(
    int sc_rule_type,
    UF_SC_input_data_p_t sc_rule
)
```

int	**sc_rule_type**
UF_SC_input_data_p_t	**sc_rule**

UF_MODL_free_section_data (view source)

Defined in: uf_sc.h

Overview
Free the UF_SC_section_data.

Environment
Internal and External

See Also
UF_MODL_create_section ,
UF_MODL_ask_section

Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

int UF_MODL_free_section_data
(
    UF_SC_section_data_p_t section
)

UF_SC_section_data_p_t	**section**

UF_MODL_init_sc_input_data (view source)

Defined in: uf_sc.h

Overview
Initiate UF_SC_input_data_t

You need to call this function if you define this data to ensure that your UF
code will never have compile problems in future versions when we could add new
parameters to the data structure to enhance the functionality.

We will set all the parameters to be zero, false, or NULL except tolerances which
will use the modeling tolerances.

Environment
Internal and External

See Also
UF_MODL_free_sc_input_data

History
This function was originally released in NX2.

Required License(s)
gateway

void UF_MODL_init_sc_input_data
(
    int rule_type,
    UF_SC_input_data_p_t rule
)

int	**rule_type**	Input	The input data type
UF_SC_input_data_p_t	**rule**	Input / Output	pointer to UF_SC_input_data_t structure

---

UF_MODL_initialize_section_data (view source)

Defined in: uf_sc.h

Overview
Initialize the UF_SC_section_data.

Environment
Internal and External

See Also
UF_MODL_create_section ,
UF_MODL_ask_section ,
UF_MODL_free_section_data
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
gateway

```
void UF_MODL_initialize_section_data
(
    UF_SC_section_data_p_t section
)
```

UF_SC_section_data_p_t	**section**

UF_MODL_remove_rules_from_face_container (view source)

Defined in: uf_sc.h

Overview
Remove existing rules from a face container. If one of the input rules is not in the container, an error will be returned.

Environment
Internal and External

See Also
UF_MODL_create_smart_face_container ,
UF_MODL_ask_rules_to_face_container ,
UF_MODL_add_rules_from_face_container
Example: ufd_modl_face_containers.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_remove_rules_from_face_container
(
    int n_rules,
    int * rule_types,
    UF_SC_input_data_t * rules,
    tag_t face_container_tag
)
```

int	**n_rules**	Input	The number of face container rules to be added.
int *	**rule_types**	Input	The types of face container rules to be added. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t *	**rules**	Input	The face container rules to be added. The face container rule data are defined in uf_sc_types.h.
tag_t	**face_container_tag**	Input	The tag of the face container.

UF_MODL_remove_rules_from_section (view source)

Defined in: uf_sc.h

Overview
Remove section rules from an existing section. If one of the input rules is not in the section, an error will be returned. The removed rules should be obtained from UF_MODL_ask_section.

Environment
Internal and External

See Also
UF_MODL_create_section ,
UF_MODL_ask_section ,
UF_MODL_add_rules_to_section ,
UF_MODL_set_start_and_direction_to_section
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_remove_rules_from_section
(
    int n_rules,
    UF_SC_section_data_t * rules,
    tag_t section_tag
)
```

int	**n_rules**	Input	The number of section builder rules.

UF_SC_section_data_t *	**rules**	Input	An array of section builder rules. The section builder rule data are defined in uf_sc_types.h.
tag_t	**section_tag**	Input	The tag of the section builder.

UF_MODL_remove_rules_from_wireframe_container (view source)

Defined in: uf_sc.h

Overview
Remove existing rules from a wireframe container. If one of the input rules is not in the container, an error will be returned.

Environment
Internal and External

See Also
UF_MODL_create_smart_wireframe_container ,
UF_MODL_ask_rules_to_wireframe_container ,
UF_MODL_add_rules_from_wireframe_container
Example: ufd_modl_wireframe_containers.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_remove_rules_from_wireframe_container
(
    int n_rules,
    int * rule_types,
    UF_SC_input_data_t * rules,
    tag_t wireframe_container_tag
)
```

int	**n_rules**	Input	The number of wireframe container rules to be added.
int *	**rule_types**	Input	The types of wireframe container rules to be added. The valid types are defined in uf_sc_types.h.
UF_SC_input_data_t *	**rules**	Input	The wireframe container rules to be added. The wireframe container rule data are defined in uf_sc_types.h.
tag_t	**wireframe_container_tag**	Input	The tag of the wireframe container.

UF_MODL_set_start_and_direction_to_section (view source)

Defined in: uf_sc.h

Overview
Set the starting object and the direction of a section.

Environment
Internal and External

See Also
UF_MODL_create_section ,
UF_MODL_ask_section ,
UF_MODL_add_rules_to_section ,
UF_MODL_remove_rules_from_section
Example: ufd_modl_sections.c

History
This function was originally released in NX2.

Required License(s)
solid_modeling

```
int UF_MODL_set_start_and_direction_to_section
(
    tag_t starting_object,
    double starting_point [ 3 ] ,
    double direction [ 3 ] ,
    tag_t section_tag
)
```

tag_t	**starting_object**	Input	The object, curve or edge, specifying the starting of the section. If the section is a loop, then the loop is started by starting_object and the direction is followed the input direction. If the section is not a loop, then the section can be started by other object, but the direction of the section is defined by the starting_object and direction.
double	**starting_point [3]**	Input	The origin of the vector defined the direction of the section. This point has to be on the starting object.
double	**direction [3]**	Input	The direction of the vector defined the direction of the section.
tag_t	**section_tag**	Input	The tag of the section builder.