uc5007 (view source)

Defined in: uf_layer.h

Overview

uc5007 create layer category -- replaced by UF_LAYER_create_category

Required License(s)

gateway

```
int uc5007
(
    int * ip1,
    const char * cp2,
    int ip3
)
```

int *	**ip1**	Input
const char *	**cp2**	
int	**ip3**	

---

uc5008 (view source)

Defined in: uf_layer.h

Overview

uc5008 read category layer -- replaced by UF_LAYER_ask_category_info

Required License(s)

gateway

```
int uc5008
(
    const char * cp1,
    int ip2,
    int * ir3
)
```

const char *	**cp1**	Input
int	**ip2**	Input
int *	**ir3**	

---

uc5009 (view source)

Defined in: uf_layer.h

Overview

uc5009 edit layer category -- replaced UF_LAYER_edit_category_layer

Required License(s)

gateway

```
int uc5009
(
    int * ip1,
    const char * cp2,
    int ip3
)
```

int *	**ip1**	Input	
const char *	**cp2**		
int	**ip3**		

UF_LAYER_ask_category_info (view source)

Defined in: uf_layer.h

Overview

Reads category name, member layers, and description.

Environment

Internal and External

See Also

UF_LAYER_category_info_p_t

Required License(s)

gateway

```
int UF_LAYER_ask_category_info
(
    tag_t category,
    UF_LAYER_category_info_p_t category_info
)
```

tag_t	**category**	Input	Category object identifier
UF_LAYER_category_info_p_t	**category_info**	Output	Pointer to category information for the given category.

UF_LAYER_ask_category_tag (view source)

Defined in: uf_layer.h

Overview
Finds the tag of a category given the category name

Environment
Internal and External

History
New in V18.0

Required License(s)
gateway

int UF_LAYER_ask_category_tag
(
const char * category_name,
tag_t * category
)

const char *	**category_name**	Input	Name of the category
tag_t *	**category**	Output	Category object identifier. If the category name does not exist, category will be NULL_TAG (but no error code will be returned).

UF_LAYER_ask_status (view source)

Defined in: uf_layer.h

Overview
Reads layer status.

Environment
Internal and External

Required License(s)
gateway

int UF_LAYER_ask_status
(
const int layer_number,
int * layer_status
)

const int	**layer_number**	Input	Layer number

int *	**layer_status**	Output	Layer status UF_LAYER_WORK_LAYER UF_LAYER_ACTIVE_LAYER UF_LAYER_REFERENCE_LAYER UF_LAYER_INACTIVE_LAYER

UF_LAYER_ask_work_layer (view source)

Defined in: uf_layer.h

Overview
Reads work layer.

Environment
Internal and External

History
New in V16.0

Required License(s)
gateway

```
int UF_LAYER_ask_work_layer
(
    int * layer_number
)
```

int *	**layer_number**	Output	Layer number

UF_LAYER_create_category (view source)

Defined in: uf_layer.h

Overview
Create a category name, member layers, and description.

Environment
Internal and External

See Also
UF_LAYER_category_info_p_t
For example please refer to example

Required License(s)
gateway

```
int UF_LAYER_create_category
(
    UF_LAYER_category_info_p_t category_info,
    tag_t * category
```

)

UF_LAYER_category_info_p_t	**category_info**	Input	Pointer to category info structure
tag_t *	**category**	Output	Category object identifier

UF_LAYER_cycle_by_layer (view source)

Defined in: uf_layer.h

Overview

Cycles the work part by layer.

First call: Returns first object in first enabled layer.
Next call: Returns next object in next enabled layer.
Last call: When all objects have been exhausted,
object_tag = NULL_TAG is returned.

Do not attempt to delete objects when cycling the database in a loop. Problems can occur when trying to read the next object when the current object has been deleted. To delete objects, save an array with the objects in it, and then when you have completed cycling, use UF_OBJ_delete_array_of_objects to delete the saved array of objects.

UF_LAYER_cycle_by_layer returns all objects on the given layer. This includes objects which are not counted as objects on the layer by the "Layer Settings" dialog.

Environment

Internal and External

Required License(s)

gateway

```
int UF_LAYER_cycle_by_layer
(
    int layer_number,
    tag_t * object_tag
)
```

int	**layer_number**	Input	Layer number to cycle, pass in a layer number of 0 to cycle all enabled layers.
tag_t *	**object_tag**	Input / Output	On input the object found by the last call to this routine. Begin the cycle by passing in object = NULL_TAG On output, the next object on the specified layer or layers. Outputs a NULL_TAG when the cycle is finished.

UF_LAYER_edit_category_descr (view source)

Defined in: uf_layer.h

Overview

Edit a category description.

Environment

Internal and External

Required License(s)

gateway

```
int UF_LAYER_edit_category_descr
(
    tag_t category,
    const char * cat_descr
)
```

tag_t	**category**	Input	Category object identifier
const char *	**cat_descr**	Input	New description for this category.

UF_LAYER_edit_category_layer (view source)

Defined in: uf_layer.h

Overview

Edit the layers associated with a category.

Environment

Internal and External

Required License(s)

gateway

```
int UF_LAYER_edit_category_layer
(
    tag_t category,
    logical layer_mask [ UF_LAYER_MAX_LAYER ]
)
```

tag_t	**category**	Input	Category object identifier
logical	**layer_mask [UF_LAYER_MAX_LAYER]**	Input	A logical for each layer, does it belong to this category or not. layer_mask[0] is TRUE if layer 1 belongs to the category, otherwise it is false. layer_mask[1] applies to layer 2, and so on.

UF_LAYER_edit_category_name (view source)

Defined in: uf_layer.h

Overview

Edit a category name.

Environment

Internal and External

Required License(s)

gateway

```
int UF_LAYER_edit_category_name
(
    tag_t category,
    const char * cat_name
)
```

tag_t	**category**	Input	Category object identifier
const char *	**cat_name**	Input	New category name

UF_LAYER_set_all_but_work (view source)

Defined in: uf_layer.h

Overview

Sets the status of all layers, except the work layer, as specified.

Environment

Internal and External

Required License(s)

gateway

```
int UF_LAYER_set_all_but_work
(
    const int layer_status
)
```

const int	**layer_status**	Input	Layer status UF_LAYER_WORK_LAYER UF_LAYER_ACTIVE_LAYER UF_LAYER_REFERENCE_LAYER UF_LAYER_INACTIVE_LAYER

UF_LAYER_set_many_layers_status (view source)

Defined in: uf_layer.h

Overview

Sets the specified layers to the corresponding specified status. The work layer is not made reference or inactive. Only one layer can be the work layer. If any error occurs, then none of the layers status is modified.

Environment
Internal and External

See Also
UF_LAYER_status_info_p_t

Required License(s)
gateway

```
int UF_LAYER_set_many_layers_status
(
    const int count_of_layers,
    UF_LAYER_status_info_p_t changes
)
```

const int	**count_of_layers**	Input	Count of layers specified
UF_LAYER_status_info_p_t	**changes**	Input	An array of structures where each element in the array is a structure that contains a layer number and status.

UF_LAYER_set_status (view source)

Defined in: uf_layer.h

Overview
Sets the layer status to either: work layer, active layer, reference layer, or inactive layer.

The status of the current work layer may not be changed. You must first set another layer to be the work layer, then change the status of the prior work layer.

Environment
Internal and External

History
V18.0 Disallow changing the status of the current work layer

Required License(s)
gateway

```
int UF_LAYER_set_status
(
    const int layer_number,
    const int layer_status
)
```

const int	**layer_number**	Input	Layer number

const int	**layer_status**	Input	Layer status UF_LAYER_WORK_LAYER UF_LAYER_ACTIVE_LAYER UF_LAYER_REFERENCE_LAYER UF_LAYER_INACTIVE_LAYER