UF_GROUP_add_member_to_group (view source)

Defined in: uf_group.h

Overview
Adds a member to a group.

Environment
Internal and External

Required License(s)
gateway

```
int UF_GROUP_add_member_to_group
(
    tag_t member_tag,
    tag_t group_tag
)
```

tag_t	**member_tag**	Input	Member object identifier to be added to the group.
tag_t	**group_tag**	Input	Group object identifier

UF_GROUP_ask_all_owning_groups (view source)

Defined in: uf_group.h

Overview
Queries the owning groups of the given tag and their count.
The function outputs the count of owning groups and the array that contains the tags of owning groups

Environment
Internal and External

History
Released in NX3

Required License(s)
gateway

```
int UF_GROUP_ask_all_owning_groups
(
    tag_t member_tag,
    int * num_owning_groups,
    tag_p_t * owning_groups_p
)
```

tag_t	**member_tag**	Input	the tag of the member whose owning groups are queried
int *	**num_owning_groups**	Output	The count of owning groups

tag_p_t *	**owning_groups_p**	Output to UF_*free*	The array that contains the tags of owning groups

UF_GROUP_ask_group_data (view source)

Defined in: uf_group.h

Overview
Queries the members of a group. After you are done using group_members, deallocate the memory using UF_free.

Environment
Internal and External

Required License(s)
gateway

```
int UF_GROUP_ask_group_data
(
    const tag_t group_tag,
    tag_t * * group_members,
    int * count_of_members
)
```

const tag_t	**group_tag**	Input	The tag of the input group
tag_t * *	**group_members**	Output to UF_*free*	An array which contains the tags of the group members. If the number of the group members is 0, this will be a NULL_TAG. Use UF_free(group_members) to free memory.
int *	**count_of_members**	Output	number of members in the group

UF_GROUP_ask_group_of_tag (view source)

Defined in: uf_group.h

Overview
Queries the tag of the group to which the specified input tag belongs.
If the input tag belongs to more than one group, the first group tag found will be returned. Use UF_GROUP_ask_all_owning_groups to see all groups an entity belongs to.

Environment
Internal and External

See Also
UF_GROUP_ask_all_owning_groups

Required License(s)

gateway

```
int UF_GROUP_ask_group_of_tag
(
    tag_t tag_of_interest,
    tag_t * group_tag
)
```

tag_t	**tag_of_interest**	Input	The object identifier of the object to check.
tag_t *	**group_tag**	Output	The tag of the group to which the tag_of_interest belongs. Returns a NULL_TAG if the tag_of_interest does not belong to any group.

UF_GROUP_create_group (view source)

Defined in: uf_group.h

Overview

Create a group from a set of group members.

Note: A call to UF_MODL_update may be necessary after creating
a group for that group to show up correctly in the Part Navigator.

Environment

Internal and External

Required License(s)

gateway

```
int UF_GROUP_create_group
(
    tag_t * group_members,
    const int count_of_members,
    tag_t * group_tag
)
```

tag_t *	**group_members**	Input	count_of_members An array which contains the tags of the group members
const int	**count_of_members**	Input	number of group members in the group
tag_t *	**group_tag**	Output	the tag of created group

UF_GROUP_del_member_from_group (view source)

Defined in: uf_group.h

Overview

Delete a member from a group.
If Part Navigator is open, deletion wont be reflected immediately.
Please Exit and re-open Part Navigator to see latest change.

Environment
Internal and External

See Also
UF_GROUP_del_member_with_refresh

Required License(s)
gateway

```
int UF_GROUP_del_member_from_group
(
    tag_t member_tag,
    tag_t group_tag
)
```

tag_t	**member_tag**	Input	Member object identifier to be deleted.
tag_t	**group_tag**	Input	group object identifier This may be NULL_TAG, which means that the member_tag should be deleted from whatever group it is a member of, if any. If group_tag is NULL_TAG and member_tag does not belong to any group, no error is returned. But if group_tag is not null, an error is returned if member_tag was not a member of group_tag before the call.

UF_GROUP_del_member_with_refresh (view source)

Defined in: uf_group.h

Overview
Delete a member from a group and refresh the Part Navigator immediately.
This call may be slow due to refresh of Part Navigator.

Environment
Internal and External

See Also
UF_GROUP_del_member_from_group

Required License(s)
gateway

```
int UF_GROUP_del_member_with_refresh
(
    tag_t member_tag,
    tag_t group_tag
)
```

tag_t	**member_tag**	Input	Member object identifier to be deleted.

tag_t	**group_tag**	Input	group object identifier This may be NULL_TAG, which means that the member_tag should be deleted from whatever group it is a member of, if any. If group_tag is NULL_TAG and member_tag does not belong to any group, no error is returned. But if group_tag is not null, an error is returned if member_tag was not a member of group_tag before the call.

UF_GROUP_is_unique_membership_group (view source)

Defined in: uf_group.h

Overview
Queries whether the group whose tag is passed as input, is UMG(Unique Member Group).
outputs TRUE - if the group is UMG and
outputs FALSE - if the group is non-UMG

Environment
Internal and External

History
Released in NX3

Required License(s)
gateway

```
int UF_GROUP_is_unique_membership_group
(
    tag_t group_tag,
    logical * is_UMG
)
```

tag_t	**group_tag**	Input	the tag of the group to be checked if it is UMG(Unique Member Group)
logical *	**is_UMG**	Output	TRUE - if the group is UMG FALSE - if the group is non-UMG

UF_GROUP_set_non_unique_membership (view source)

Defined in: uf_group.h

Overview
Sets the group whose tag is passed as input, as non-UMG (Non -Unique Member Group) i.e. its members can be
members of any other groups (UMG or non-UMG).

Environment
Internal and External

History
Released in NX3

Required License(s)

gateway

```
int UF_GROUP_set_non_unique_membership
(
    tag_t group_tag
)
```

tag_t	**group_tag**	Input	the tag of the group to be set as non_UMG.

UF_GROUP_set_unique_membership (view source)

Defined in: uf_group.h

Overview

Sets the group whose tag is passed as input, as UMG (Unique Member Group)i.e. its members can be
members of only other non-UMG groups only.

Environment

Internal and External

History

Released in NX3

Required License(s)

gateway

```
int UF_GROUP_set_unique_membership
(
    tag_t group_tag
)
```

tag_t	**group_tag**	Input	the tag of the group to be set as UMG(Unique Member Group).

UF_GROUP_ungroup_all (view source)

Defined in: uf_group.h

Overview

Remove all members of a group, deletes the group, and ungroups the
immediate group and all subgroups. If the group specified is a
subgroup, all of its members become members of the ancestor group.

Environment

Internal and External

Required License(s)

gateway

```
int UF_GROUP_ungroup_all
(
    tag_t group_tag
)
```

tag_t	**group_tag**	Input	Group object identifier

UF_GROUP_ungroup_top (view source)

Defined in: uf_group.h

Overview

Remove all members of a group, deletes the group, ungroups only the immediate group, and allows subgroups to remain.
If the group specified is a subgroup, all of its members become members of the ancestor group.

Environment

Internal and External

Required License(s)

gateway

```
int UF_GROUP_ungroup_top
(
    tag_t group_tag
)
```

tag_t	**group_tag**	Input	Group object identifier of the group to be ungrouped.
