UF_LIB_ask_lib_attribute_values (view source)

Defined in: uf_lib.h

Overview

This function provides a list of attribute values of an NX object that was previously retrieved from a library. The library is referenced by the specified UF_CAM_db_object_t. The values are the current values (in the library) of the specified attributes of the NX object.

The caller should use UF_free_string_array to free these returned arrays.

Environment

Internal and External

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_ask_lib_attribute_values
(
    UF_CAM_db_object_t db,
    tag_t ug_object,
    int count,
    const char * * db_alias,
    char * * * values
)
```

UF_CAM_db_object_t	**db**	Input	- the db object to query in
tag_t	**ug_object**	Input	- the NX object to query about
int	**count**	Input	- the number of attribute to ask their values of
const char * *	**db_alias**	Input	- the list of attribute db aliases to ask the values of
char * * *	**values**	Output to UF_*free*	- the requested values. This is an array of strings that must be freed by calling UF_free_string_array.

UF_LIB_ask_libref (view source)

Defined in: uf_lib.h

Overview

This function provides the library reference (or libref) of the specified NX object. The libref is an attribute which is returned for every record in every record set. It is also the attribute which is used to retrieve an object from a library. An NX object will only have a libref if it was created by being retrieved from a library.

Environment

Internal and External

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_ask_libref
(
    tag_t ug_object,
    const char * * libref
)
```

tag_t	**ug_object**	Input	- the desired NX object
const char * *	**libref**	Output to UF_*free*	- the libref of the specified object

UF_LIB_ask_record_in_rset (view source)

Defined in: uf_lib.h

Overview

This function provides a record from a record set. The record is returned as an array of strings. Each element of the array is the value of an attribute of the specified record. The ith element of the array is the value of the ith attribute returned from ask_rset_map. Note that all values are returned as strings. If the value has type double or int the caller will need to convert as required by the application.

The caller should use UF_free_string_array to free these returned arrays.

Environment

Internal and External

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_ask_record_in_rset
(
    UF_RSET_t rset,
    int record_num,
    int * count,
    const char * * * values
)
```

UF_RSET_t	**rset**	Input	- the desired record set
int	**record_num**	Input	- the desired record number (0 based)

int *	**count**	Output	- the number of values of attributes
const char * * *	**values**	Output to UF_*free*	- the values of the attributes. The caller should call UF_free_string_array to free this array.

UF_LIB_ask_rset_count (view source)

Defined in: uf_lib.h

Overview

This function returns the number of records that the specified UF_RSET_t contains.

Environment

Internal and External

History

Released in V17.0

Required License(s)

gateway

```
int UF_LIB_ask_rset_count
(
    UF_RSET_t rset,
    int * count
)
```

UF_RSET_t	**rset**	Input	- the UF_RSET_t whose record count is desired
int *	**count**	Output	- the number of records in the UF_RSET_t

UF_LIB_ask_rset_map (view source)

Defined in: uf_lib.h

Overview

This function provides a description (or map) of the contents of a record set. It provides a list of the attributes that each record in the record set has as well as the type of each of those attributes. One can think of a record set as a table. The rows of the table are the individual records in the record set. The columns headings are the attributes that are found in the record set.
NOTE: Use UF_LIB_free_rset_map to free returned memory
See the description

Environment

Internal and External

See Also

UF_LIB_free_rset_map

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_ask_rset_map
(
    UF_RSET_t rset,
    int * count,
    const char * * * attr_names,
    const char * * attr_types
)
```

UF_RSET_t	**rset**	Input	- the record set of interest. This was returned by execute_query.
int *	**count**	Output	- the number of attributes in the record set.
const char * * *	**attr_names**	Output to UF_*free*	count - the name of the attributes in the record set.
const char * *	**attr_types**	Output to UF_*free*	count - the types of the attributes. The ith attr_types is the type of the ith attr_names.

UF_LIB_delete_rset (view source)

Defined in: uf_lib.h

Overview

This function deletes the specified UF_RSET_t.

Environment

Internal and External

History

Released in V17.0

Required License(s)

gateway

```
int UF_LIB_delete_rset
(
    UF_RSET_t rset
)
```

UF_RSET_t	**rset**	Input	- the record set to delete

UF_LIB_execute_query (view source)

Defined in: uf_lib.h

Overview

This function will execute a query in a library and return a record set containing the objects in the library that satisfy the query.
See the above File Description for a discussion of the query syntax. The Class Query in the Definition File is &&ed to the query specified here as a parameter. The contents of the RSET returned is determined by the RSET definition for the specified class name. This definition is found in the Definition File.

Environment

Internal and External

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_execute_query
(
    UF_CAM_db_object_t db,
    const char * cls_name,
    const char * query,
    int * count,
    UF_RSET_t * rset
)
```

UF_CAM_db_object_t	**db**	Input	- the library object in which to execute the query.
const char *	**cls_name**	Input	- the class name
const char *	**query**	Input	- the query
int *	**count**	Output	- the number of objects satisfying the query
UF_RSET_t *	**rset**	Output	- the record of objects that satisfy the query

---

UF_LIB_execute_query_for_count (view source)

Defined in: uf_lib.h

Overview

This function will execute a query in a library and return a count of the number of objects in the library that satisfy the query. See the above File Description for a discussion of the query syntax. The Class Query in the Definition File is &&ed to the query specified here as a parameter.

Environment

Internal and External

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_execute_query_for_count
(
    UF_CAM_db_object_t db,
    const char * cls_name,
    const char * query,
    int * count
)
```

UF_CAM_db_object_t	**db**	Input	- the library object in which to execute the query
const char *	**cls_name**	Input	- the class name
const char *	**query**	Input	- the query
int *	**count**	Output	- the number of objects in the library that satisfy the query.

UF_LIB_free_rset_map (view source)

Defined in: uf_lib.h

Overview

This function frees the memory returned from UF_LIB_ask_rset_map

Environment

Internal and External

See Also

UF_LIB_ask_rset_map

History

Released in V16.0

Required License(s)

gateway

```
int UF_LIB_free_rset_map
(
    int count,
    const char * * attr_names,
    const char * attr_types
)
```

int	**count**	Input	- the number of attributes in the record set.

const char * *	**attr_names**	Input	- the name of the attributes in the record set.
const char *	**attr_types**	Input	- the types of the attributes. The ith attr_types is the type of the ith attr_names.

UF_LIB_merge_rsets (view source)

Defined in: uf_lib.h

Overview

This function merges the contents of two record sets. The merged record set contains the records of 'rset1' followed by the records of 'rset2'. The records appear in the same order they did in their original record sets. 'rset1' and 'rset2' are unchanged.

Environment

Internal and External

History

Released in V17.0

Required License(s)

gateway

```
int UF_LIB_merge_rsets
(
    UF_RSET_t rset1,
    UF_RSET_t rset2,
    UF_RSET_t * merged_rset
)
```

UF_RSET_t	**rset1**	Input	- the record set to merge
UF_RSET_t	**rset2**	Input	- the other record to merge
UF_RSET_t *	**merged_rset**	Output	- the merged record set

UF_LIB_sort_rset (view source)

Defined in: uf_lib.h

Overview

This function sorts the contents of a record set. The sort key is of the form:

db_alias1/{ a|d },db_alias2/{ a|d },...,db_aliasN/{ a|d }

where a = ascending, d = descending. For example, if Diameter and Cost are both DB Aliases that are found in the record set one could sort the set primarily by Diameter in ascending order and then, within Diameter, sort by Cost in decreasing order with a sort key of:

Diameter/a,Cost/d

Here Diameter is the primary key. Records of the set that have the same Diameter are then sorted by Cost. The original record set is not changed.

Environment

Internal and External

History

Released in V17.0

Required License(s)

gateway

```
int UF_LIB_sort_rset
(
    UF_RSET_t rset,
    const char * sort_key,
    UF_RSET_t * sorted_rset
)
```

UF_RSET_t	**rset**	Input	- the record set to sort
const char *	**sort_key**	Input	- the sort key to sort by
UF_RSET_t *	**sorted_rset**	Output	- the sorted record set