

UF_KF_add_search_library [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Add a path to the search path for classes.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_add_search_library
(  
    const char * path_name  
)
```

const char *	path_name	Input	Path name of directory that contains the .dfa files. This will be added to the current search path for .dfa files.
--------------	------------------	-------	--

UF_KF_adopt_nx_objects [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Adopt NX objects into the Knowledge Fusion world

If NX classes are enabled, then an you can determine which type of instance you will adopt by using the `nx_type` argument. If `nx_type` is set to `TRUE`, then you will adopt an NX instance. If `nx_type` is set to `false`, then you will adopt an NX instance.

However, if NX classes are disabled, then you can only adopt NX instances, regardless of the setting of `nx_type`.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

```
UF_KF_set_nx_classes_enabled
UF_KF_ask_nx_classes_enabled
UF_KF_adopt_ug_object
```

History

released in NX5.0

Required License(s)

gateway

```
int UF_KF_adopt_nx_objects
(
    tag_t * ug_objects,
    int number_of_objects,
    logical nx_type,
    UF_KF_instance_p_t ** instances
)
```

<code>tag_t *</code>	<code>ug_objects</code>	Input	array of tags of NX objects to be adopted into Fusion
<code>int</code>	<code>number_of_objects</code>	Input	number of objects in the above array
<code>logical</code>	<code>nx_type</code>	Input	= TRUE - adopt as NX instance object = FALSE - adopt as NX instance object
<code>UF_KF_instance_p_t *</code>	<code>instances</code>	Output to UF_*free*	array of adopted instances. Free with UF_free.

UF_KF_adopt_ug_objects [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Adopt NX objects into the Knowledge Fusion world

If NX classes are enabled, then an instance of an NX object will be adopted. Otherwise, an instance of an NX object will be adopted. For example, if you are adopting and instance of a block feature, you will either get an NX_block or an NX_block instance.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

```
UF_KF_set_nx_classes_enabled
UF_KF_ask_nx_classes_enabled
```

History

Originally release in v18.0

Required License(s)

gateway

```
int UF_KF_adopt_ug_objects
(
    tag_t * ug_objects,
    int number_of_objects,
    UF_KF_instance_p_t ** instances
)
```

tag_t *	ug_objects	Input	array of tags of NX objects to be adopted into Fusion
int	number_of_objects	Input	number of objects in the above array
UF_KF_instance_p_t **	instances	Output to UF_*free*	array of adopted instances. Free with UF_free.

UF_KF_ask_all_classes (view source)

Defined in: uf_kf.h

Overview

Return a sorted array of all classes currently known by NX KF.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

[UF_KF_ask_classes](#)

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_all_classes
(
    int * num_classes,
    char *** classes
)
```

int *	num_classes	Output	Number of class names found
-------	-------------	--------	-----------------------------

char * * *	classes	Output to UF_*free*	num_classes Array of class names. Free with UF_free_string_array.
------------	----------------	---------------------	--

UF_KF_ask_ancestor_classes [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the component classes of a given class.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_ancestor_classes
(
    const char * class_name,
    int * n_ancestors,
    char * * * ancestor_names
)
```

const char *	class_name	Input	name of the class for which components are asked
int *	n_ancestors	Output	Number of components found in the class
char * * *	ancestor_names	Output to UF_*free*	Array of component names found in the class. Free with UF_free_string_array.

UF_KF_ask_base_unit_of_measure [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the base unit of a measure. For example, "mm" will be returned for "Length" in a metric part. The empty string will be returned for an invalid unit.

Return

Return code :
= 0 : successful

> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in NX301

Required License(s)

gateway

```
int UF_KF_ask_base_unit_of_measure
(
    const char * measure,
    char ** base_unit
)
```

const char *	measure	Input	Measure of value
char **	base_unit	Output to UF_*free*	Base unit of measure Free with UF_free()

UF_KF_ask_boolean [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the value of a Boolean Knowledge Fusion value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_boolean
(
    UF_KF_value_p_t value,
    logical * data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
-----------------	--------------	-------	------------------------

logical *	data	Output	Data of value
-----------	------	--------	---------------

UF_KF_ask_classes (view source)

Defined in: uf_kf.h

Overview

Return a sorted array of classes. Use the filter option to filter the type of classes desired.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

[UF_KF_ask_all_classes](#)

History

Originally released in v18.0

Required License(s)

gateway

```
int UF_KF_ask_classes
(
    UF_KF_class_type_t filter_choice,
    int * num_classes,
    char * * * classes
)
```

UF_KF_class_type_t	filter_choice	Input	filter choice: UF_KF_ASK_USER_ONLY UF_KF_ASK_SYSTEM_ONLY UF_KF_ASK_ALL
int *	num_classes	Output	Number of class names found
char * * *	classes	Output to UF_*free*	Array of class names. Free with UF_free_string_array

UF_KF_ask_default_formula (view source)

Defined in: uf_kf.h

Overview

Get the default formula for a rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_default_formula
(
    UF_KF_rule_p_t rule,
    char ** formula
)
```

UF_KF_rule_p_t	rule	Input	Rule object
char **	formula	Output to UF_*free*	Text of formula. May be NULL if no default exists. Free with UF_free.

UF_KF_ask_error [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the value of an Error from Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_error
(
    UF_KF_value_p_t value,
    int * error
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
int *	error	Output	Data of value

UF_KF_ask_frame (view source)

Defined in: uf_kf.h

Overview

Get the value of a Frame Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V18

Required License(s)

gateway

```
int UF_KF_ask_frame
(
    UF_KF_value_p_t value,
    double data [ 16 ]
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
double	data [16]	Output	Data of value. A UF_MTX4 transformation matrix. All the UF_MTX4 functions apply.

UF_KF_ask_fusion_object (view source)

Defined in: uf_kf.h

Overview

Get a fusion object from an NX object. Returns a NULL_TAG when the NX object has no associated Knowledge Fusion instance object.

Note: For some Knowledge Fusion instance types the associated NX object will be a feature not an object. For example, to get a ug_block type Knowledge Fusion instance, you would need to pass the BLOCK feature's tag_t not the resulting UF_solid_type/UF_solid_body_subtype object's tag_t.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_fusion_object
(
    tag_t ug_object,
    UF_KF_instance_p_t * instance
)
```

tag_t	ug_object	Input	NX object
UF_KF_instance_p_t *	instance	Output	Knowledge Fusion instance object



UF_KF_ask_instance [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the value of an Instance Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_instance
(
    UF_KF_value_p_t value,
    UF_KF_instance_p_t * instance
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
UF_KF_instance_p_t *	instance	Output	Data of value

UF_KF_ask_instance_class [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the class for an instance.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_instance_class
(
    const char * name_chain,
    char ** class_name
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
char **	class_name	Output to UF_*free*	Name of the class that this instance uses. Free with UF_free

UF_KF_ask_instance_of_args [\(view source\)](#)

Defined in: uf_kf.h

Overview

Return the calling instance of Defun.

Return

0 if function succeeds, otherwise
error code (non-zero).

Environment

Internal and External

History

Released in V18.1

Required License(s)

gateway

```
int UF_KF_ask_instance_of_args
(
    UF_KF_value_p_t * values,
    int n_values,
    UF_KF_instance_p_t * instance
)
```

UF_KF_value_p_t *	values	Input	Argument values of Defun
int	n_values	Input	Number of argument values
UF_KF_instance_p_t *	instance	Output	Instance calling Defun

UF_KF_ask_integer [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the value of a Integer Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_integer
(
    UF_KF_value_p_t value,
    int * data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
int *	data	Output	Data of value

UF_KF_ask_list [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the value of a List Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_list
(
    UF_KF_value_p_t value,
    UF_KF_list_p_t * list
)
```

<code>UF_KF_value_p_t</code>	value	Input	Knowledge Fusion value
<code>UF_KF_list_p_t *</code>	list	Output to UF_*free*	Knowledge Fusion list Free with UF_KF_free_list_object_contents.

UF_KF_ask_list_count [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Count the number of elements in a list

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

[UF_KF_ask_list_item](#)

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_list_count
(
    UF_KF_list_p_t list,
    int * count
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list
int *	count	Output	Number of elements in the list

UF_KF_ask_list_item [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Ask the value of an item in a list object.

Return

int - Error code indicating success (0) or failure (<>0)

Environment

Internal and External

See Also

[UF_KF_ask_list_count](#)

History

Originally release in v18.0

Required License(s)

gateway

```
int UF_KF_ask_list_item
(
    UF_KF_list_p_t list,
    int index,
    UF_KF_value_p_t * value
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list
int	index	Input	Index of item in the list. (1 is the first item)
UF_KF_value_p_t *	value	Output to UF_*free*	The value of the list item. Free with UF_KF_free_rule_value.

UF_KF_ask_list_of_instance [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the list object from an instance that is a list datatype.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v17.0

Required License(s)

gateway

```
int UF_KF_ask_list_of_instance
(
    const char * name_chain,
    UF_KF_value_p_t * list_object
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
UF_KF_value_p_t *	list_object	Output to UF_*free*	The list value. Free with UF_KF_free_rule_value.

UF_KF_ask_measure_of_value [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the measure of a Number Knowledge Fusion structure. This function returns strings like "Length", "Area" and "Volume". The empty string is returned for number values that do not have a measure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in NX301

Required License(s)
gateway

```
int UF_KF_ask_measure_of_value
(
    UF_KF_value_p_t value,
    const char ** measure
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion data structure
const char **	measure	Output	Measure of value

UF_KF_ask_name (view source)

Defined in: uf_kf.h

Overview
Get the value of a Name Knowledge Fusion value.

Return
Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment
Internal and External

History
Released in V18

Required License(s)
gateway

```
int UF_KF_ask_name
(
    UF_KF_value_p_t value,
    unsigned int * data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
unsigned int *	data	Output	Data of value

UF_KF_ask_name_chain_of_instance (view source)

Defined in: uf_kf.h

Overview

Ask the name chain of a Knowledge Fusion instance

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v18.0
Modified in NX 11 user is now responsible for freeing the returned string

Required License(s)

gateway

```
int UF_KF_ask_name_chain_of_instance
(
    UF_KF_instance_p_t instance,
    char ** name_chain
)
```

UF_KF_instance_p_t	instance	Input	Knowledge Fusion instance
char **	name_chain	Output to UF_*free*	Name chain of instance. Free with UF_free.

UF_KF_ask_name_of_string (view source)

Defined in: uf_kf.h

Overview

Get the name representation in KF of an input string. For example determine the name in KF for the boolean name "Unite".

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

- UF_KF_ask_string_of_name
- UF_KF_ask_name
- UF_KF_make_name

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_name_of_string
(
    const char * string,
    unsigned int * name
)
```

const char *	string	Input	Name string
unsigned int *	name	Output	Knowledge Fusion name of the string

UF_KF_ask_number [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the value of a Number Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_number
(
    UF_KF_value_p_t value,
    double * data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion data structure
double *	data	Output	Data of value

UF_KF_ask_nx_classes_enabled [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Return true if NX classes are enabled for the Create Child rule dialog and for adoption in the specified part.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

History

Released in NX400

Required License(s)

gateway

```
int UF_KF_ask_nx_classes_enabled
(
    const tag_t object_in_part,
    logical * nx_classes_enabled
)
```

<code>const tag_t</code>	<code>object_in_part</code>	Input	Tag of object in part
<code>logical *</code>	<code>nx_classes_enabled</code>	Output	True --> nx classes enabled

UF_KF_ask_parameter_formula [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get formula of a given parameter rule.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_parameter_formula
(
    const char * name_chain,
    const char * param_name,
    char ** formula
)
```

)

const char *	name_chain	Input	Reference chain of instance, with trailing colon
const char *	param_name	Input	Name of the parameter
char **	formula	Output to UF_*free*	Formula for the rule. Free with UF_free.

UF_KF_ask_parameter_type [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get datatype of a given parameter rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_parameter_type
(
    const char * name_chain,
    const char * param_name,
    char ** data_type
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
const char *	param_name	Input	Name of the parameter
char **	data_type	Output to UF_*free*	Data type of the parameter. Free with UF_free.

UF_KF_ask_parameters [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get list of parameters that have been specified for an instance. The formula for each individual parameter can be obtained by calling UF_KF_ask_parameter_formula.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_parameters
(
    const char * name_chain,
    char * * * parameter_names,
    int * num_params
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
char * * *	parameter_names	Output to UF_*free*	Array of parameter names that were specified. Free with UF_free_string_array.
int *	num_params	Output	Number of parameters found

UF_KF_ask_point [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the value of a Point Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_point
(
    UF_KF_value_p_t value,
    double data [ 3 ]
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
double	data [3]	Output	Data of value

UF_KF_ask_rule_name (view source)

Defined in: uf_kf.h

Overview

Get the name of a rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rule_name
(
    UF_KF_rule_p_t rule,
    char ** rule_name
)
```

UF_KF_rule_p_t	rule	Input	Rule object
char **	rule_name	Output to UF_*free*	Rule name string. Free with UF_free.

UF_KF_ask_rule_of_instance (view source)

Defined in: uf_kf.h

Overview

Find the rule object of a given name in an instance.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rule_of_instance
(
    const char * name_chain,
    const char * rule_name,
    UF_KF_rule_p_t * rule
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
const char *	rule_name	Input	Name of rule
UF_KF_rule_p_t *	rule	Output	Found rule object

UF_KF_ask_rule_of_name [\(view source\)](#)

Defined in: uf_kf.h

Overview

Find the rule object for a name in a class.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rule_of_name
(
    const char * class_name,
    const char * rule_name,
    UF_KF_rule_p_t * rule
)
```

const char *	class_name	Input	Name of class to find rule in.
const char *	rule_name	Input	Name of rule to find.
UF_KF_rule_p_t *	rule	Output	Found rule object.

UF_KF_ask_rule_text_of_referencing_object [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Return rule text for referencing an NX object in Knowledge Fusion. Note that the rule text may reference the container object rather than the given object. For example, the rule text may reference the owning CSYS feature rather than the given datum plane or datum axis. If the rule text of the given object is desired then call in the NXOpen API the method `GetObjectText` which is on the class `RuleManager`.

Return

0 if function succeeds, otherwise error code (non-zero).

Environment

Internal and External

History

Released in V18

Required License(s)

gateway

```
int UF_KF_ask_rule_text_of_referencing_object
(
    tag_t ug_object_tag,
    char ** rule_text
)
```

<code>tag_t</code>	ug_object_tag	Input	Tag of the NX object
<code>char **</code>	rule_text	Output to <code>UF_*free*</code>	Rule text to be used in Knowledge Fusion. Free with <code>UF_free</code> .

UF_KF_ask_rule_type [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the data type of a rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rule_type
(
    UF_KF_rule_p_t rule,
    char ** rule_type
)
```

UF_KF_rule_p_t	rule	Input	Rule object
char **	rule_type	Output to UF_*free*	Rule type string. Free with UF_free.

UF_KF_ask_rule_types [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get list of data types that rules can be.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rule_types
(
    int * num_types,
    char *** data_types
)
```

int *	num_types	Output	Number of rule types existing
-------	-----------	--------	-------------------------------

char * * *	data_types	Output to UF_*free*	Array of data type names. Free with UF_free_string_array.
------------	-------------------	---------------------	--

UF_KF_ask_rules [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

get the names of all the rules under a given instance.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rules
(
    const char * name_chain,
    int * num_rules,
    char * * * rules
)
```

const char *	name_chain	Input	Reference chain of the instance
int *	num_rules	Output	Number of rules found for this instance
char * * *	rules	Output to UF_*free*	Array of rules found under the instance. Free with UF_free_string_array.

UF_KF_ask_rules_of_class [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get all the rules in a class.

Return

Return code :
= 0 : successful
> 0 : failing error number

< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_rules_of_class
(
    const char * class_name,
    int * num_rules,
    UF_KF_rule_p_t ** rules
)
```

const char *	class_name	Input	name of the class of which rules are to be found
int *	num_rules	Output	Number of rules found in the class
UF_KF_rule_p_t **	rules	Output to UF_*free*	Array of rules found in the class. Free with UF_free.

UF_KF_ask_string (view source)

Defined in: uf_kf.h

Overview

Get the value of a String Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17
Modified in NX 11 user is now responsible for freeing the returned string

Required License(s)

gateway

```
int UF_KF_ask_string
(
    UF_KF_value_p_t value,
    char ** string
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
char **	string	Output to UF_*free*	Value. Must be freed.

UF_KF_ask_string_of_name (view source)

Defined in: uf_kf.h

Overview

Get the string representation of a name in KF.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

See Also

[UF_KF_ask_name_of_string](#)
[UF_KF_ask_name](#)
[UF_KF_make_name](#)

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_string_of_name
(
    unsigned int name,
    char ** string
)
```

unsigned int	name	Input	Knowledge Fusion name
char **	string	Output to UF_*free*	Name string. Free with UF_free.

UF_KF_ask_tag (view source)

Defined in: uf_kf.h

Overview

Get the value of a Hostpointer Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_tag
(
    UF_KF_value_p_t value,
    tag_t * data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
tag_t *	data	Output	Data of value

UF_KF_ask_type [\(view source\)](#)

Defined in: uf_kf.h

Overview

Get the type of a Knowledge Fusion structure.

The supported types are defined in UF_KF_type_t

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_type
(
    UF_KF_value_p_t data,
    UF_KF_type_t * type
)
```

UF_KF_value_p_t	data	Input	Knowledge Fusion data structure
UF_KF_type_t *	type	Output	Value type

UF_KF_ask_ug_object (view source)

Defined in: uf_kf.h

Overview

Get the NX object from a Knowledge Fusion object

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_ug_object
(
    UF_KF_instance_p_t instance,
    tag_t * ug_object
)
```

UF_KF_instance_p_t	instance	Input	Knowledge Fusion instance structure
tag_t *	ug_object	Output	NX object

UF_KF_ask_user (view source)

Defined in: uf_kf.h

Overview

Get the value of a User Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V18

Required License(s)

gateway

```
int UF_KF_ask_user
(
    UF_KF_value_p_t value,
    void ** data
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
void **	data	Output	Data of value. Free if necessary.

UF_KF_ask_user_class_dir (view source)

Defined in: uf_kf.h

Overview

Return the user class directory.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_ask_user_class_dir
(
    char ** dir
)
```

char **	dir	Output to UF_*free*	User class directory path. Free with UF_free.
---------	-----	---------------------	--

UF_KF_ask_value_of_list_item (view source)

Defined in: `uf_kf.h`

Overview

Ask the value of an item in a list object.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v17.0

Required License(s)

gateway

```
int UF_KF_ask_value_of_list_item
(
    UF_KF_value_p_t list_obj,
    int list_index,
    char ** value_str,
    char ** datatype,
    UF_KF_value_p_t * sub_list_obj
)
```

UF_KF_value_p_t	list_obj	Input	List value
int	list_index	Input	Index of item in the list. (1 is the first item)
char * *	value_str	Output to UF_*free*	Text of item's value. Free with UF_free.
char * *	datatype	Output to UF_*free*	Data type of item. Free with UF_free
UF_KF_value_p_t *	sub_list_obj	Output to UF_*free*	If this item is a list value, then this parameter contains the list value. If this item is not a list value, then this parameter will be NULL. Free with UF_KF_free_rule_value.

UF_KF_ask_vector [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the value of a Vector Knowledge Fusion structure.

Return

Return code :
= 0 : successful
> 0 : failing error number

< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_ask_vector
(
    UF_KF_value_p_t value,
    double data [ 3 ]
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion value
double	data [3]	Output	Data of value

UF_KF_count_list_items [\(view source\)](#)

Defined in: uf_kf.h

Overview

Count the number of items in a list object.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v17.0 will be obsoleted after v19 use
UF_KF_ask_list_count

Required License(s)

gateway

```
int UF_KF_count_list_items
(
    UF_KF_list_p_t list,
    int * n_items
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list
int *	n_items	Output	number of items in list object.

UF_KF_create_child_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create or modify a child rule for a given instance. If the rule already exists then replace the existing rule. The input parameters will be formatted into an Knowledge Fusion rule as follows: "(rule_type) rule_name: rule_text;" This routine should be used only to create/edit child rules. This routine does not call UF_MODL_update so the user needs to. Note that child rules can only be created on the "Root:"

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_create_child_rule
(
    tag_t part_of_rule,
    const char * name_chain,
    const char * rule_name,
    const char * child_class_name,
    int num_parameters,
    char ** parameter_names,
    char ** parameter_rules,
    char ** generated_name
)
```

<code>tag_t</code>	<code>part_of_rule</code>	Input	Part to create or modify rule in
<code>const char *</code>	<code>name_chain</code>	Input	Reference chain of instance, with trailing colon, where rule will be created or modified. If create then must be "Root:" or NULL. Child rules can only be created on the "Root:"
<code>const char *</code>	<code>rule_name</code>	Input	Name of rule.
<code>const char *</code>	<code>child_class_name</code>	Input	Class of instance
<code>int</code>	<code>num_parameters</code>	Input	number of parameters in the parameter_names array

char * *	parameter_names	Input	num_parameters array of names of parameters specified by the user.
char * *	parameter_rules	Input	num_parameters array of rules for the parameters whose names are in "parameter_names" above. Each rule may or may not have a trailing semicolon.
char * *	generated_name	Output to UF_*free*	name generated for the child object. Currently always returns NULL.

UF_KF_create_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create or modify a rule for a given instance. If the rule already exists then replace the existing rule. The input parameters will be formatted into an Knowledge Fusion rule as follows: "(rule_type) rule_name: rule_text;" This routine can create/modify both child rules and attribite rules. This routine calls UF_MODL_update for the user. UF_KF_create_rule_no_update() does not. Note that child rules can only be created on the "Root:"

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_create_rule
(
    const char * name_chain,
    const char * rule_name,
    const char * rule_type,
    const char * rule_text,
    const char * leading_comment
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon, where rule will be created or modified
const char *	rule_name	Input	Name of rule
const char *	rule_type	Input	Type of rule. Type names may be obtained by calling UF_KF_ask_rule_types. This can also

contain behavior flags, so "Number Parameter" is a valid rule_type, as is "Child". If create "Child" then must be "Root:" or NULL. Child rules can only be created on the "Root:"

const char *	rule_text	Input	Text for rule. For an instance rule, this must be enclosed in curly braces. For an attribute rule, just the formula
const char *	leading_comment	Input	Leading comment for rule. OPTIONAL-can be null. If non-null, make sure that the string is preceded by the pound sign '#' and followed by a new-line character '\n' Example: "#Leading Comment\n"

UF_KF_create_rule_no_update [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create or modify a rule for a given instance. If the rule already exists then replace the existing rule. The input parameters will be formatted into an Knowledge Fusion rule as follows: "(rule_type) rule_name: rule_text;" This routine can create/modify both child rules and attribute rules. This routine does not call UF_MODL_update so the user needs to. Note that child rules can only be created on the "Root:"

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v18.01

Required License(s)

gateway

```
int UF_KF_create_rule_no_update
(
    const char * name_chain,
    const char * rule_name,
    const char * rule_type,
    const char * rule_text,
    const char * leading_comment
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon, where rule will be created or modified
const char *	rule_name	Input	Name of rule

const char *	rule_type	Input	Type of rule. Type names may be obtained by calling UF_KF_ask_rule_types. This can also contain behavior flags, so "Number Parameter" is a valid rule_type, as is "Child". If create "Child" then must be "Root:" or NULL. Child rules can only be created on the "Root:"
const char *	rule_text	Input	Text for rule. For an instance rule, this must must be enclosed in curly braces. For a attribute rule, just the formula
const char *	leading_comment	Input	Leading comment for rule. OPTIONAL-can be null.

UF_KF_delete_class_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Delete a rule from a given class.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_delete_class_rule
(
    const char * class_name,
    const char * rule_name
)
```

const char *	class_name	Input	Class that rule is part of
const char *	rule_name	Input	Name of rule

UF_KF_delete_instance_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Delete a rule from a given instance.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_delete_instance_rule
(
    const char * name_chain,
    const char * rule_name
)
```

const char *	name_chain	Input	Instance that rule is part of.
const char *	rule_name	Input	Name of rule

UF_KF_evaluate_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Force a rule to be evaluated.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_evaluate_rule
(
    const char * name_chain,
    UF_KF_value_p_t * value
)
```

const char *	name_chain	Input	Reference chain for the rule
UF_KF_value_p_t *	value	Output to UF_*free*	Value of the rule after evaluation. Free with UF_KF_free_rule_value

UF_KF_evaluate_rule_to_string [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Force a rule to be evaluated and return the result as a printable string.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_evaluate_rule_to_string
(
    const char * name_chain,
    char ** value
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
char **	value	Output to UF_*free*	Value of the rule after evaluation. Free with UF_free

UF_KF_free_list_object_contents [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Free the contents of the list object returned from UF_KF_ask_list.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally release in v17.0

Required License(s)

gateway

```
int UF_KF_free_list_object_contents
(
    UF_KF_list_p_t list
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list
----------------	------	-------	-----------------------

UF_KF_free_rule_value [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Free a value returned by UF_KF_evaluate_rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_free_rule_value
(
    UF_KF_value_p_t value
)
```

UF_KF_value_p_t	value	Input	value to free returned by UF_KF_evaluate_rule
-----------------	-------	-------	---

UF_KF_init_part [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Initialize a part class.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

See Also

[UF_KF_is_initialized](#)

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_init_part
(
    tag_t part
)
```

tag_t part Input Tag of the part to initialize
--

UF_KF_is_cached_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is a cached rule.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_cached_rule
(
    UF_KF_rule_p_t rule,
    logical * cached
)
```


UF_KF_rule_p_t	rule	Input	Rule object
logical *	cached	Output	TRUE if the rule is a cached rule, else FALSE

UF_KF_is_canonical_rule (view source)

Defined in: uf_kf.h

Overview

Check whether a rule is a canonical rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_canonical_rule
(
    UF_KF_rule_p_t rule,
    logical * canonical
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	canonical	Output	TRUE if the rule is a canonical rule, else FALSE

UF_KF_is_child_list_instance (view source)

Defined in: uf_kf.h

Overview

Find out if the instance is a child list.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_child_list_instance
(
    const char * name_chain,
    logical * is_list_instance
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
logical *	is_list_instance	Output	TRUE if the instance is a child list, else FALSE

UF_KF_is_child_list_rule [\(view source\)](#)

Defined in: uf_kf.h

Overview

Check whether a rule is a child list rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_child_list_rule
(
    UF_KF_rule_p_t rule,
    logical * is_child_list
)
```

UF_KF_rule_p_t	rule	Input	Rule object returned from UF_KF_ask_rules_of_class
logical *	is_child_list	Output	TRUE if the rule is a child list rule, else FALSE

UF_KF_is_child_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is a child rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_child_rule
(
    UF_KF_rule_p_t rule,
    logical * child
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	child	Output	TRUE if the rule is a child rule, else FALSE

UF_KF_is_dynamic [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is dynamic.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_dynamic
(
    UF_KF_rule_p_t rule,
    logical * dynamic
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	dynamic	Output	TRUE if the rule is dynamic, else FALSE

UF_KF_is_hidden_rule (view source)

Defined in: uf_kf.h

Overview

Check whether a rule is a hidden rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_hidden_rule
(
    UF_KF_rule_p_t rule,
    logical * hidden
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	hidden	Output	TRUE if the rule is a hidden rule, else FALSE

UF_KF_is_initialized (view source)

Defined in: uf_kf.h

Overview

This routine will verify if KF is initialized in the part and its availability. If initializing KF in the part succeeds then this function will return TRUE.

Return

int - Error code indicating success (0) or failure (<>0)

Environment

Internal and External

See Also

[UF_KF_init_part](#)

History

Originally released in v18.0

Required License(s)

gateway

```
int UF_KF_is_initialized
(
    logical * is_initialized
)
```

logical *	is_initialized	Output	TRUE if KF is initialized and is available, else FALSE
-----------	-----------------------	--------	--

UF_KF_is_local [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is local (defined on this class, not on an ancestor).

Return

Return code :

- = 0 : successful
- > 0 : failing error number
- < 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_local
(
    const char * class_name,
    UF_KF_rule_p_t rule,
```

logical * local
)

const char *	class_name	Input	The name of the class in question
UF_KF_rule_p_t	rule	Input	Rule object
logical *	local	Output	TRUE if the rule is a local rule, else FALSE

UF_KF_is_lookup_rule ([view source](#))

Defined in: `uf_kf.h`

Overview

Check whether a rule is a lookup rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_lookup_rule
(
    UF_KF_rule_p_t rule,
    logical * lookup
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	lookup	Output	TRUE if the rule is a lookup rule, else FALSE

UF_KF_is_method ([view source](#))

Defined in: `uf_kf.h`

Overview

Check whether a rule is a method.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_method
(
    UF_KF_rule_p_t rule,
    logical * method
)
```

UF_KF_rule_p_t	rule	Input	Rule object
logical *	method	Output	TRUE if the rule is a method, else FALSE

UF_KF_is_modifiable_rule (view source)

Defined in: uf_kf.h

Overview

Check whether a rule is a modifiable rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_modifiable_rule
(
    UF_KF_rule_p_t rule,
    logical * modifiable
)
```

UF_KF_rule_p_t	rule	Input	Rule object
----------------	------	-------	-------------

<code>logical *</code>	modifiable	Output	TRUE if the rule is a modifiable rule, else FALSE
------------------------	-------------------	--------	---

UF_KF_is_parameter_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is a parameter rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_parameter_rule
(
    UF_KF_rule_p_t rule,
    logical * parameter
)
```

<code>UF_KF_rule_p_t</code>	rule	Input	Rule object
<code>logical *</code>	parameter	Output	TRUE if the rule is a parameter rule, else FALSE

UF_KF_is_rule_computed [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Ask whether a rule has been evaluated yet.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_rule_computed
(
    const char * name_chain,
    const char * rule_name,
    logical * evaluated
)
```

const char *	name_chain	Input	Reference chain of instance, with trailing colon
const char *	rule_name	Input	Name of the rule
logical *	evaluated	Output	TRUE if it has been evaluated, FALSE otherwise.

UF_KF_is_rule_in_class [\(view source\)](#)

Defined in: uf_kf.h

Overview

Check whether a given rule exists for a class.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_rule_in_class
(
    const char * class_name,
    const char * rule_name,
    logical * rule_in_class
)
```

const char *	class_name	Input	Name of class to check for rule
const char *	rule_name	Input	Name of rule to check for

<code>logical *</code>	<code>rule_in_class</code>	Output	TRUE if the rule is in the class, else FALSE
------------------------	----------------------------	--------	---

UF_KF_is_uncached_rule [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Check whether a rule is an uncached rule.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_is_uncached_rule
(
    UF_KF_rule_p_t rule,
    logical * uncached
)
```

<code>UF_KF_rule_p_t</code>	<code>rule</code>	Input	Rule object
<code>logical *</code>	<code>uncached</code>	Output	TRUE if the rule is an uncached rule, else FALSE

UF_KF_list_pop [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Get the next element from the list. This routine will return the next list element, it will not remove the pop element from the list. You must remember to free the head of the list since pop moves down the list to the tail.

```
UF_KF_ask_list ( list_value, &list ); // Get list
next = list; // Save next element
UF_KF_list_pop ( next, &value1, &next ); // Get value from list
UF_KF_list_pop ( next, &value2, &next ); // Get value from list
UF_KF_free_list_object_contents ( list ); // Free list when finished
// Use the newly created values ...
UF_KF_free_rule_value ( value1 ); // Free value when finished
```

```
UF_KF_free_rule_value ( value2 ); // Free value when finished
```

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_list_pop
(
    UF_KF_list_p_t list,
    UF_KF_value_p_t * element,
    UF_KF_list_p_t * next
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list
UF_KF_value_p_t *	element	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value.
UF_KF_list_p_t *	next	Output	Next on Knowledge Fusion list

UF_KF_list_push [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Put the element at the top of the list. This routine will return the new list element, it will add the push element to the list. You must remember to free the head of the list since push moves up the list to the new head.

```
UF_KF_ask_list ( list_value, &list ); // Get list
UF_KF_list_push ( list, value1, &list ); // Add value to list
UF_KF_list_push ( list, value2, &list ); // Add value to list
UF_KF_free_rule_value ( value1 ); // Free value when finished
UF_KF_free_rule_value ( value2 ); // Free value when finished
// Use the newly created list ...
UF_KF_free_list_object_contents ( list ); // Free list when finished
```

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_list_push
(
    UF_KF_list_p_t list,
    UF_KF_value_p_t element,
    UF_KF_list_p_t * next
)
```

UF_KF_list_p_t	list	Input	Knowledge Fusion list.
UF_KF_value_p_t	element	Input	Knowledge Fusion value to push onto the list
UF_KF_list_p_t *	next	Output	Next on Knowledge Fusion list

UF_KF_make_boolean [\(view source\)](#)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion Boolean value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_boolean
(
    logical data,
    UF_KF_value_p_t * value
)
```

logical	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_error (view source)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion Error value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_error
(
    int data,
    UF_KF_value_p_t * value
)
```

int	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_frame (view source)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion Frame value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V18

Required License(s)

gateway

```
int UF_KF_make_frame
(
    double data [ 16 ],
    UF_KF_value_p_t * value
)
```

double	data [16]	Input	Data of value. A UF_MTX4 transformation matrix. All the UF_MTX4 functions apply.
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_instance [\(view source\)](#)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion instance value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_instance
(
    UF_KF_instance_p_t data,
    UF_KF_value_p_t * value
)
```

UF_KF_instance_p_t	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_integer [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion Integer value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_integer
(
    int data,
    UF_KF_value_p_t * value
)
```

int	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_list [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion List value.

The input to create a list is a Knowledge Fusion value which can be created using the following routines:
UF_KF_make_string,
UF_KF_make_tag,
UF_KF_make_vector,
UF_KF_make_point ...

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_list
(
    UF_KF_list_p_t data,
    UF_KF_value_p_t * value
)
```

UF_KF_list_p_t	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value.

UF_KF_make_name (view source)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion Name value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V18

Required License(s)

gateway

```
int UF_KF_make_name
(
    int data,
    UF_KF_value_p_t * value
)
```

int	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_number [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion Number value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_number
(
    double data,
    UF_KF_value_p_t * value
)
```

double	data	Input	Data of value
<code>UF_KF_value_p_t *</code>	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_point [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion Point value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_point
(
    double data [ 3 ],
    UF_KF_value_p_t * value
)
```

double	data [3]	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_string (view source)

Defined in: uf_kf.h

Overview

Create a Knowledge Fusion String value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_string
(
    const char * data,
    UF_KF_value_p_t * value
)
```

const char *	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_tag [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion Hostpointer value. This is a way to reference an NX object without adopting it to Knowledge Fusion.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V17

Required License(s)

gateway

```
int UF_KF_make_tag
(
    tag_t data,
    UF_KF_value_p_t * value
)
```

<code>tag_t</code>	data	Input	Data of value
<code>UF_KF_value_p_t *</code>	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_user [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Create a Knowledge Fusion User value.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in V18

Required License(s)
gateway

```
int UF_KF_make_user
(
    void * data,
    UF_KF_value_p_t * value
)
```

void *	data	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_make_vector [\(view source\)](#)

Defined in: uf_kf.h

Overview
Create a Knowledge Fusion Vector value.

Return
Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment
Internal and External

History
Released in V17

Required License(s)
gateway

```
int UF_KF_make_vector
(
    double data [ 3 ],
    UF_KF_value_p_t * value
)
```

double	data [3]	Input	Data of value
UF_KF_value_p_t *	value	Output to UF_*free*	Knowledge Fusion value. Free with UF_KF_free_rule_value. Do not free if being returned from a CFunc.

UF_KF_remove_all_rules [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Remove all rules in the current part.

Return

0 if function succeeds, otherwise
error code (non-zero).

Environment

Internal and External

History

Released in NX204

Required License(s)

gateway

```
int UF_KF_remove_all_rules
(  
    void  
)
```

UF_KF_remove_rule_only [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Remove a dynamic child or attribute rule, but leave the geometry.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in NX 3.0.4

Required License(s)

gateway

```
int UF_KF_remove_rule_only
(  
    const char * name_chain,  
    const char * rule_name  
)
```

const char *	name_chain	Input	Instance that rule is part of.
--------------	-------------------	-------	--------------------------------

const char *	rule_name	Input	Name of rule
--------------	------------------	-------	--------------

UF_KF_revert [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Re-read the Knowledge Fusion language description of all loaded classes from the text files. The filenames are not needed, as the Knowledge Fusion search rules are used to find them.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_revert
(
    const char * class_name
)
```

const char *	class_name	Input	If it is not NULL, it is the class_name whose design needs to reloaded. If it is NULL, all the existing classes in the system need to reload.
--------------	-------------------	-------	---

UF_KF_set_measure_of_value [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Set the measure of a Number Knowledge Fusion structure. This function accepts strings like "Length", "Area" and "Volume". The empty string is accepted for number values that do not have a measure.

Return

Return code :
 = 0 : successful
 > 0 : failing error number
 < 0 : failing error number

Environment

Internal and External

History

Released in NX301

Required License(s)

gateway

```
int UF_KF_set_measure_of_value
(
    UF_KF_value_p_t value,
    const char * measure
)
```

UF_KF_value_p_t	value	Input	Knowledge Fusion data structure
const char *	measure	Input	Measure of value

UF_KF_set_nx_classes_enabled (view source)

Defined in: uf_kf.h

Overview

If true then set NX classes enabled for the Create Child rule dialog and for adoption in the specified part.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Released in NX400

Required License(s)

gateway

```
int UF_KF_set_nx_classes_enabled
(
    const tag_t object_in_part,
    logical nx_classes_enabled
)
```

const tag_t	object_in_part	Input	Tag of object in part
logical	nx_classes_enabled	Input	True --> nx classes enabled

UF_KF_set_user_class_dir [\(view source\)](#)

Defined in: `uf_kf.h`

Overview

Set the user class dir.

Return

Return code :
= 0 : successful
> 0 : failing error number
< 0 : failing error number

Environment

Internal and External

History

Originally released in v17.0

Required License(s)

gateway

```
int UF_KF_set_user_class_dir
(
    char ** new_dirs,
    int num_dirs
)
```

char **	new_dirs	Input	. Array of directories to set as user class dir
int	num_dirs	Input	Number of directories

UF_KF_UGMGR_ask_user_dfa_classes [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function gets the user defined KF classes stored in the Teamcenter Engineering database. If there is more than one folder having the same name then all the KF classes defined in those folders are returned.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_ask_user_dfa_classes
(
```



```
const char* folder_name,  
char* * * user_dfa_classes,  
int* num_classes  
)
```

const char*	folder_name	Input	Name of the folder
char* * *	user_dfa_classes	Output to UF_*free*	list of KF classes in the specified folder. The list should be freed by calling UF_free_string_array
int*	num_classes	Output	number of KF classes

UF_KF_UGMGR_ask_user_dfa_funcs [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function gets the user defined KF functions stored in the Teamcenter Engineering database. If there is more than one folder having the same name then all the KF funcs defined in those folders are returned.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_ask_user_dfa_funcs  
(  
    const char* folder_name,  
    char* * * user_dfa_funcs,  
    int* num_funcs  
)
```

const char*	folder_name	Input	Name of the folder
char* * *	user_dfa_funcs	Output to UF_*free*	list of KF functions file in the specified folder. The list should be freed by calling UF_free_string_array
int*	num_funcs	Output	number of KF funcs

UF_KF_UGMGR_initialize_export [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function sets the options for export of DFA files in command line mode

Environment

Internal and External

History

Originally released in NX2

Required License(s)

gateway

```
int UF_KF_UGMGR_initialize_export
(
    int option
)
```

int	option	Input	Use UF_KF_UGMGR_EXPORT_DFA_IN_PART to export DFA files in individual part Use UF_KF_UGMGR_EXPORT_DFA_IN_ASSEMBLY to export DFA files used by entire assembly
-----	---------------	-------	---

UF_KF_UGMGR_initialize_import [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function sets the options for import of DFA files in command line mode

Environment

Internal and External

History

Originally released in NX2

Required License(s)

gateway

```
int UF_KF_UGMGR_initialize_import
(
    logical is_dfa_list
)
```

logical	is_dfa_list	Input	True if input is a file containing the names of files to import. If true the value of the -part switch is a file containing a list of DFA files to import (1 per line). Otherwise the value of the -part switch is assumed to be the DFA file to import.
---------	--------------------	-------	--

UF_KF_UGMGR_is_folder_exists [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function checks whether the specified folder exists in Teamcenter Engineering.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_is_folder_exists
(
    const char* folder_name,
    logical* folder_exists
)
```

const char*	folder_name	Input	Name of the folder to search for in the Teamcenter Engineering Database.
logical*	folder_exists	Output	TRUE if exists, else FALSE

UF_KF_UGMGR_is_item_dfa_type [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function checks whether the item is DFA

Environment

Internal and External

History

Originally released in NX2

Required License(s)

gateway

```
int UF_KF_UGMGR_is_item_dfa_type
(
    char* item_name,
    logical* dfa_type
)
```

char*	item_name	Input	Item Name
logical*	dfa_type	Output	TRUE = Item is a DFA else FALSE

UF_KF_UGMGR_is_user_dfa_class_exists [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function checks whether the specified KF class exists in the Teamcenter Engineering database.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_is_user_dfa_class_exists
(
    const char* dfa_class_name,
    logical* class_exists
)
```

const char*	dfa_class_name	Input	Name of the KF class to search for
logical*	class_exists	Output	True if exists, else FALSE

UF_KF_UGMGR_is_user_dfa_func_exists [\(view source\)](#)

Defined in: `uf_kf_ugmgr.h`

Overview

This function checks whether the specified KF function exists in the Teamcenter Engineering database.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_is_user_dfa_func_exists
(
    const char* dfa_func_name,
    logical* func_exists
)
```

const char*	dfa_func_name	Input	Name of the KF functions to search for
logical*	func_exists	Output	True if exists, else FALSE

UF_KF_UGMGR_list_user_classes_referenced

([view source](#))

Defined in: `uf_kf_ugmgr.h`

Overview

This function gets all the user KF classes referenced in the part file.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_list_user_classes_referenced
(
    tag_t part_tag,
    char* ** user_classes_list,
    int* num_classes
)
```

tag_t	part_tag	Input	tag of the part
char* **	user_classes_list	Output to UF_*free*	List of classes referenced by the part specified. This list has to be freed by calling UF_free_string_array
int*	num_classes	Output	Number of KF classes referenced in the part file

UF_KF_UGMGR_list_user_funcs_referenced

([view source](#))

Defined in: `uf_kf_ugmgr.h`

Overview

This function gets all the user KF funcs referenced in the part file.

This function always returns 0 functions as references to functions are not maintained in Teamcenter.

Environment

Internal and External

History

Originally released in V18.0

Required License(s)

gateway

```
int UF_KF_UGMGR_list_user_funcs_referenced
(
    tag_t part_tag,
    char* ** user_funcs_list,
    int* num_funcs
)
```

tag_t	part_tag	Input	tag of the part
char* **	user_funcs_list	Output to UF_*free*	List of functions referenced by the part specified. This list has to be freed by calling UF_free_string_array
int*	num_funcs	Output	Number of KF functions referenced in the part file

UF_UGMGR_KF_export_dfa_in_part (view source)

Defined in: uf_kf_ugmgr.h

Overview

This function exports DFA classes and functions that are referenced in a part file

Environment

Internal and External

History

Originally released in NX2

Required License(s)

gateway

```
int UF_UGMGR_KF_export_dfa_in_part
(
    const char* part_name
)
```

const char*	part_name	Input	CLI Name of part
-------------	-----------	-------	------------------