UF_SIM_activate_tool (view source)

Defined in: uf_sim_commands.h

Overview
Activate a mounted tool.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_activate_tool
(
    UF_SIM_engine_p_t engine,
    char* tool_name
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
char*	**tool_name**	Input	- the name of the tool to activate

UF_SIM_ask_axis_dof_junction (view source)

Defined in: uf_sim_commands.h

Overview
Returns the full junction name (= <comp_name>@<junction_name> of the axis that defines the degree of freedom.
Also, returns the direction of the degree of freedom axis. The jct_name character array has to be of length 2UF_OBJ_NAME_BUFSIZE.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_axis_dof_junction
(
    UF_SIM_engine_p_t engine,
    const char* axis,
    char jct_name [ 2*(UF_OBJ_NAME_BUFSIZE
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**axis**	Input	- the name of the axis
char	**jct_name [2*(UF_OBJ_NAME_BUFSIZE**		

UF_SIM_ask_axis_is_reversal_allowed (view source)

Defined in: uf_sim_commands.h

Overview
Ask if the reversal mode for the given NC axis is allowed.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_axis_is_reversal_allowed
(
    UF_SIM_engine_p_t engine,
    const char* axis_name,
    logical * allow_reversal
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**axis_name**	Input	- name of the NC axis
logical *	**allow_reversal**	Output	- switch of axis

UF_SIM_ask_axis_limits (view source)

Defined in: uf_sim_commands.h

Overview
Returns the axis limits.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_axis_limits
(
    UF_SIM_engine_p_t engine,
    const char* axis,
    double* min,
    double* max
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

const char*	**axis**	Input	- the name of the axis
double*	**min**	Output	- the minimum limit of the axis
double*	**max**	Output	- the maximum limit of the axis

UF_SIM_ask_axis_position (view source)

Defined in: uf_sim_commands.h

Overview

Return the current axis position. Axis can be linear or rotary. The current position is the position of the axis after all input events prior to this event have been processed. Coordinates are returned in axis coordinates.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_ask_axis_position
(
    UF_SIM_engine_p_t engine,
    const char * axis,
    double * position
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char *	**axis**	Input	- the name of the axis
double *	**position**	Output	- the current axis position

UF_SIM_ask_axis_rotary_dir_mode (view source)

Defined in: uf_sim_commands.h

Overview

Return the direction of rotation mode for the specified NC axis. This setting controls the path (CW or CCW) a rotary axis will follow when it goes to target position. The possible rotation modes are:

UF_SIM_AXIS_ROT_MAGNITUDE_DETERMINES_DIRECTION
The rotary axis behaves like a linear axis where if target position > 0 then it rotate in CCW. If < 0 then it rotates CW. For example if current position is 355 and target position is 370 then it rotates in CCW 15 degrees. If target position is 10 then it rotates in CW 345 degrees.

UF_SIM_AXIS_ROT_ALWAYS_SHORTEST

Moves to the target position in shortest path where possible. This is the default behavior.

UF_SIM_AXIS_ROT_SIGN_DETERMINES_DIRECTION
For positive target position the axis rotates CCW if negative it rotates CW.

UF_SIM_AXIS_ROT_ALWAYS_CLW
Rotates in CW direction only.

UF_SIM_AXIS_ROT_ALWAYS_CCLW
Rotates in CCW direction only.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_axis_rotary_dir_mode
(
    UF_SIM_engine_p_t engine,
    const char* axis_name,
    UF_SIM_axis_rot_dir_type_t * rot_mode
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**axis_name**	Input	- name of the axis
UF_SIM_axis_rot_dir_type_t *	**rot_mode**	Output	- rot mode of the axis

---

UF_SIM_ask_comp_from_dof (view source)

Defined in: uf_sim_commands.h

Overview
Return the name of the KIM component where the given degree of freedom is designed to.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_comp_from_dof
(
    UF_SIM_engine_p_t engine,
    char* degof_name,
    char comp_name [ UF_OBJ_NAME_BUFSIZE ]
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

char*	**degof_name**	Input	- name of the degree of freedom
char	**comp_name [UF_OBJ_NAME_BUFSIZE]**	Output	- the name of the component

---

UF_SIM_ask_cording_tol (view source)

Defined in: uf_sim_commands.h

Overview
Query and return the chordal tolerance.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_cording_tol
(
    UF_SIM_engine_p_t engine,
    double* cording_tol
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
double*	**cording_tol**	Output	- the cording tolerance

UF_SIM_ask_degof_data (view source)

Defined in: uf_sim_commands.h

Overview
Query the data for a given degree of freedom.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_degof_data
(
    UF_SIM_engine_p_t engine,
    const char* degof_name,
    UF_SIM_KIM_degof_types_t* degof_type,
    double* lower_limit,
    double* upper_limit
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**degof_name**	Input	- name of the d. of freedom
UF_SIM_KIM_degof_types_t*	**degof_type**	Output	- type of the d. of freedom
double*	**lower_limit**	Output	- lower limit
double*	**upper_limit**	Output	- upper limit

UF_SIM_ask_if_degof_exists (view source)

Defined in: uf_sim_commands.h

Overview
Returns TRUE or FALSE whether the specified degree of freedom exists in the KIM model.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_if_degof_exists
(
    UF_SIM_engine_p_t engine,
    const char* degof_name,
    logical* degof_exists
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**degof_name**	Input	- name of the degree of freedom
logical*	**degof_exists**	Output	- existence of the degree of freedom

---

UF_SIM_ask_immediate_update (view source)

Defined in: uf_sim_commands.h

Overview
Returns the immediate update setting. See UF_SIM_set_immediate_update for details.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_immediate_update
(
    UF_SIM_engine_p_t engine,
    logical* immediate_update
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
logical*	**immediate_update**	Output	- return immediate update setting

---

UF_SIM_ask_init_junction_xform (view source)

Defined in: uf_sim_commands.h

Overview

Return the specified junction offset and transformation. To be compatible with csys and msys structure an array with three vectors (X,Y and Z) of the junction with respect to the ABS coordinate system is returned.
This means the matrix has to be transformed first. This is the INITIAL transformation of the junction before the machine tool starts moving.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_ask_init_junction_xform
(
    UF_SIM_engine_p_t engine,
    const char* jct_name,
    double* xval,
    double* yval,
    double* zval,
    double* matrix
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**jct_name**	Input	- name of the junction
double*	**xval**	Output	- offset in x direction
double*	**yval**	Output	- offset in y direction
double*	**zval**	Output	- offset in z direction
double*	**matrix**	Output	- matrix

UF_SIM_ask_is_junction_dependent (view source)

Defined in: uf_sim_commands.h

Overview
Returns TRUE or FALSE whether the specified junction is dependent on specified NC axis.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_is_junction_dependent
(
    UF_SIM_engine_p_t engine,
    const char* jct_name,
    const char* axis_name,
    char dependent [ 256 ]
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**jct_name**	Input	- name of the junction
const char*	**axis_name**	Input	- name of the axis
char	**dependent [256]**	Output	- dependency of the junction

UF_SIM_ask_is_junction_exist (view source)

Defined in: uf_sim_commands.h

Overview
Return whether the specified junction exists or not.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_is_junction_exist
(
    UF_SIM_engine_p_t engine,
    const char* jct_name,
    logical* jct_exists
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

const char*	**jct_name**	Input	- the name of the junction
logical*	**jct_exists**	Output	- existence of the junction

UF_SIM_ask_junction_xform (view source)

Defined in: uf_sim_commands.h

Overview

Request and return the specified junction offset and transformation.
To be compatible with csys and msys structure an array with three vectors (X,Y and Z) of the junction with respect to the ABS coordinate system is returned. This means the matrix first has to be tranformed first.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_ask_junction_xform
(
    UF_SIM_engine_p_t engine,
    const char* jct_name,
    double* xval,
    double* yval,
    double* zval,
    double* matrix
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**jct_name**	Input	- the name of the junction
double*	**xval**	Output	- offset value in x direction
double*	**yval**	Output	- offset value in y direction
double*	**zval**	Output	- offset value in z direction
double*	**matrix**	Output	- matrix between ZCS and MTCS

UF_SIM_ask_kim_comp_name_by_id (view source)

Defined in: uf_sim_commands.h

Overview

Request the name of a component of the KIM structure by a given id.

Environment

Internal and External

History
Released in NX2

int UF_SIM_ask_kim_comp_name_by_id
(
UF_SIM_engine_p_t engine,
int system_class,
char* comp_id,
char comp_name [UF_OBJ_NAME_BUFSIZE]
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
int	**system_class**	Input	- system class of the searched comp
char*	**comp_id**	Input	- the id of the component
char	**comp_name [UF_OBJ_NAME_BUFSIZE]**	Output	- name of the component

---

UF_SIM_ask_mtcs_junction (view source)

Defined in: uf_sim_commands.h

Overview
Returns the junction name that represents the machine tool coordinate system.
The jct_name character array has to be of length UF_OBJ_NAME_BUFSIZE.

Environment
Internal and External

History
Released in NX2

int UF_SIM_ask_mtcs_junction
(
UF_SIM_engine_p_t engine,
char jct_name [UF_OBJ_NAME_BUFSIZE]
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
char	**jct_name [UF_OBJ_NAME_BUFSIZE]**	Output	- the name of the queried junction

---

UF_SIM_ask_mtd_units (view source)

Defined in: uf_sim_commands.h

Overview
Return the machine tool driver units. Those will be the units to input/output data from S&V commands.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_mtd_units
(
    UF_SIM_engine_p_t engine,
    UF_SIM_unit_type_t * units
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
UF_SIM_unit_type_t *	**units**	Output	- units

UF_SIM_ask_nc_axes_of_mtool (view source)

Defined in: uf_sim_commands.h

Overview
Return the list of NC axes of the machine tool in the KIM structure.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_nc_axes_of_mtool
(
    UF_SIM_engine_p_t engine,
    int* no_of_axes,
    char axis_name_list [ 4096 ]
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
int*	**no_of_axes**	Output	- Number of NC axes
char	**axis_name_list [4096]**	Output	- list of NC axis names

---

UF_SIM_ask_sim_engine (view source)

Defined in: uf_sim_commands.h

Overview
Returns the object id of the simulation engine that is linked to MOM object.

Environment
Internal and External

History
Originally released in NX2

```
int UF_SIM_ask_sim_engine
(
    UF_MOM_id_t mom_id,
    UF_SIM_engine_p_t * engine_id
)
```

UF_MOM_id_t	**mom_id**	Input	- the MOM object ID
UF_SIM_engine_p_t *	**engine_id**	Output	- the simulation engine

---

UF_SIM_ask_status_send_nc_command_msg (view source)

Defined in: uf_sim_commands.h

Overview
Return the setting for sending NC command message to the feedback processor.
By default it is issued automatically by the system.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_ask_status_send_nc_command_msg
(
    UF_SIM_engine_p_t engine,
    logical * status
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
logical *	**status**	Output	- the status

UF_SIM_ask_tool_offsets (view source)

Defined in: uf_sim_commands.h

Overview

Request the offset values for a given tool.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_ask_tool_offsets
(
    UF_SIM_engine_p_t engine,
    char* tool_name,
    double* xval,
    double* yval,
    double* zval
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
char*	**tool_name**	Input	- the name of the tool
double*	**xval**	Output	- value in x direction from tool offset
double*	**yval**	Output	- value in y direction from tool offset
double*	**zval**	Output	- value in z direction from tool offset

---

UF_SIM_convert_nurbs_to_position_data (view source)

Defined in: uf_sim_commands.h

Overview

Calculates based on the given parameters for NURBS an array of end-points of linear segments. Based on the existing documentation of NURBS (Post) it is expected, that points are 3 dimensional and knots are one dimensional. Therefore knots has to have the length of knot_count and cntr_pnts has to have the length of 3cntr_pnt_count.

Output
point_count: number of returned points in the array
positions: list of points
size of array is positions(point_count3)
The order is point after point,
position(0) -> point-1-X
position(1) -> point-1-Y
position(2) -> point-1-Z
position(3) -> point-2-X
position(4) -> point-2-Y
...
Memory of positions is allocated inside this function and

must be freed from the caller of this function

Environment
Internal and External

History
Released in NX2

int UF_SIM_convert_nurbs_to_position_data
(
UF_SIM_engine_p_t engine,
int cntr_point_count,
int order,
int knot_count,
double* knots,
double* cntr_points,
int* point_count,
double* * positions
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
int	**cntr_point_count**	Input	- number of control points
int	**order**	Input	- order of the nurbs
int	**knot_count**	Input	- number of knot vectors
double*	**knots**	Input	- array of knot vectors
double*	**cntr_points**	Input	- array of control points
int*	**point_count**	Output	- number of points in the array
double* *	**positions**	Output to UF_*free*	- return array of points

---

UF_SIM_create_junction (view source)

Defined in: uf_sim_commands.h

Overview
Add new junction and attach it to specified component. Unlike some other S&V commands, the creation is done right away. No events are created or triggered to perform this action.

Environment
Internal and External

History
Released in NX2

int UF_SIM_create_junction
(

```
    UF_SIM_engine_p_t engine,
    const char* jct_name,
    const char* destination_comp,
    double origin [ 3 ] ,
    double matrix [ 9 ]
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**jct_name**	Input	- the name of the junction
const char*	**destination_comp**	Input	- the name of the dest. cmp
double	**origin [3]**	Input	- the vector of the origin
double	**matrix [9]**	Input	- the matrix of the junction

UF_SIM_dbg_end (view source)

Defined in: uf_sim_commands.h

Overview

End S&V commands debugging tools. It also closes the output listing device.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_dbg_end
(
    UF_SIM_engine_p_t engine
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

---

UF_SIM_dbg_set_output (view source)

Defined in: uf_sim_commands.h

Overview

Activate or deactivate one of the following debugging tokens that controls the output of the corresponding data to the listing device. If token not specified then debugging is turned on or off according to on_off value.

UF_SIM_DBG_OUTPUT_ERROR output warnings/errors
UF_SIM_DBG_OUTPUT_NC_CMD output the NC program (i.e. G codes)
UF_SIM_DBG_OUTPUT_CEVENT output CEvents (i.e. post events)
UF_SIM_DBG_OUTPUT_INPUT_SV_EVENT output S&V input events
UF_SIM_DBG_OUTPUT_OUTPUT_SV_EVENT output S&V output events

UF_SIM_DBG_OUTPUT_PROCESSED_SV_EVENT output S&V processed events those includes the output events and those events created by the S&V processors
UF_SIM_DBG_OUTPUT_SV_CMD output ALL S&V commands that are called
UF_SIM_DBG_OUTPUT_ALL output all the above data

All tokens must be specified in upper case.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_dbg_set_output
(
    UF_SIM_engine_p_t engine,
    logical on_off,
    const char* token
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
logical	**on_off**	Input	- switch to set the output on or off
const char*	**token**	Input	- name of the token to switch on or off

---

UF_SIM_dbg_start (view source)

Defined in: uf_sim_commands.h

Overview
Initialize S&V commands debugging tools. It also opens a listing device where output debug messages are written.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_dbg_start
(
    UF_SIM_engine_p_t engine,
    const char* title
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**title**	Input	- titel of the debug listing window

UF_SIM_dbg_write_message (view source)

Defined in: uf_sim_commands.h

Overview
Write message in the debugging output listing. If debugging is not active, no message will be written.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_dbg_write_message
(
    UF_SIM_engine_p_t engine,
    const char* msg
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**msg**	Input	- the message string, which goes to the debug output window

UF_SIM_delay (view source)

Defined in: uf_sim_commands.h

Overview
Cause the system to dwell and stop event execution for specified duration.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_delay
(
    UF_SIM_engine_p_t engine,
    const char* label,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**label**	Input	- label to identify the action

double	**duration**	Input	- the duration to stop in sec

UF_SIM_delete_junction (view source)

Defined in: uf_sim_commands.h

Overview
Delete a junction. Unlike some other S&V commands, the deletion is done right away. No events are created or triggered to perform this action.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_delete_junction
(
    UF_SIM_engine_p_t engine,
    const char* jct_name
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**jct_name**	Input	- the name of the junction

UF_SIM_dialog_set_item (view source)

Defined in: uf_sim_commands.h

Overview
Create an event to set a dialog item attributes. The item identifier is a unique identifier of dialog item. The attributes defines the keyword that identifies the attribute followed by the attribute value according to dialog item attribute syntax. E.g. LABEL=<UGT0201_013> LIST=<X,Y,Z>.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_dialog_set_item
(
    UF_SIM_engine_p_t engine,
    const char* item_id,
    const char* attributes
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**item_id**	Input	- the name of the item
const char*	**attributes**	Input	- the value of the attribute

---

UF_SIM_end_of_simulation (view source)

Defined in: uf_sim_commands.h

Overview
Indicate end of simulation.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_end_of_simulation
(
    UF_SIM_engine_p_t engine
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

---

UF_SIM_feedback_message (view source)

Defined in: uf_sim_commands.h

Overview
Issue the specified feedback message.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_feedback_message
(
    UF_SIM_engine_p_t engine,
    char* label,
    char* message
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object

char*	**label**	Input	- the type of the feedback message
char*	**message**	Input	- the information of the feedback message

---

UF_SIM_find_comp_by_name (view source)

Defined in: uf_sim_commands.h

Overview
Return true if the given component name is found in the KIM tree hierachy. Started by the given component name, false if the given name isn't found.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_find_comp_by_name
(
    UF_SIM_engine_p_t engine,
    char* start_comp,
    char* search_comp,
    logical* is_found
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
char*	**start_comp**	Input	- name of the base component
char*	**search_comp**	Input	- name of the component to search
logical*	**is_found**	Output	- result of searching

UF_SIM_mount_kim_comp (view source)

Defined in: uf_sim_commands.h

Overview
Mount a specified component (source component) with a given coordinate system (source junction) of the machine tool KIM model at a specified location (destination component and destination junction).

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_mount_kim_comp
(
    UF_SIM_engine_p_t engine,
    char* source_comp,
    char* source_jct,
    char* destination_comp,
    char* destination_jct,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
char*	**source_comp**	Input	- the name of the source component
char*	**source_jct**	Input	- the name of the source junction
char*	**destination_comp**	Input	- the name of the destination cmp
char*	**destination_jct**	Input	- the name of the destination jct
double	**duration**	Input	- the time in seconds to complete this mount on the machine tool

---

UF_SIM_mount_tool (view source)

Defined in: uf_sim_commands.h

Overview

Mount a specified tool by its tool_id on a specified component (destination component and destination junction). The tool can be specified by different classification types.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_mount_tool
(
    UF_SIM_engine_p_t engine,
    UF_SIM_tool_class_t tool_class,
    char* tool_id,
    char* destination_comp,
    char* destination_jct,
    double duration,
    char tool_name [ UF_OBJ_NAME_BUFSIZE ]
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
UF_SIM_tool_class_t	**tool_class**	Input	- tool classification identifier
char*	**tool_id**	Input	- the tool_id of the tool

char*	**destination_comp**	Input	- the name of the destin. cmp
char*	**destination_jct**	Input	- the name of the destin. jct
double	**duration**	Input	- time in seconds to complete this mount on the machine tool
char	**tool_name [UF_OBJ_NAME_BUFSIZE]**	Output	- name of the tool

---

UF_SIM_move_linear_axis (view source)

Defined in: uf_sim_commands.h

Overview
Move the specified linear axis by the specified amount.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_move_linear_axis
(
    UF_SIM_engine_p_t engine_id,
    char* axis,
    double value,
    double duration
)
```

UF_SIM_engine_p_t	**engine_id**	Input	- the simulation engine object
char*	**axis**	Input	- the name of the linear NC-axis
double	**value**	Input	- the new position of the axis
double	**duration**	Input	- the time in seconds to complete this move on the machine. This value is effected by the current feed rate.

UF_SIM_move_rotary_axis (view source)

Defined in: uf_sim_commands.h

Overview
Move the specified rotary axis by the specified amount.

Environment

Internal and External

History
Released in NX2

int UF_SIM_move_rotary_axis
(
UF_SIM_engine_p_t engine_id,
char* axis,
double value,
double duration
)

UF_SIM_engine_p_t	**engine_id**	Input	- the simulation engine object
char*	**axis**	Input	- the name of the linear NC-axis
double	**value**	Input	- the new position of the axis
double	**duration**	Input	- the time in seconds to complete this move on the machine. This value is effected by the current feed rate.

UF_SIM_msg_nc_command (view source)

Defined in: uf_sim_commands.h

Overview
Issue command messages mainly to the feedback processor to act according to the command message. SIM_msg_nc_command is a message to the feedback processor about the NC command that has been executed. The action field contains the 'G codes'.

Environment
Internal and External

History
Released in NX2

int UF_SIM_msg_nc_command
(
UF_SIM_engine_p_t engine,
const char* action
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**action**	Input	- NC command

UF_SIM_msg_program_mark (view source)

Defined in: uf_sim_commands.h

Overview
Send program mark message to message window. Next/Previous operation steps to this mark.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_msg_program_mark
(
    UF_SIM_engine_p_t engine,
    const char* program_mark
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**program_mark**	Input	- name of the program mark

---

UF_SIM_msg_user_feedback (view source)

Defined in: uf_sim_commands.h

Overview
Issue command messages mainly to the feedback processor to act according to the command message. SIM_msg_user_feedback is a message to the feedback processor to invoke the TCL procedure specified as the 'proc' argument.
Note: That the prefix MOM_ will be added to the procedure before invocation.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_msg_user_feedback
(
    UF_SIM_engine_p_t engine,
    const char* proc
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**proc**	Input	- name of the a TCL procedure

UF_SIM_mtd_init (view source)

Defined in: uf_sim_commands.h

Overview
Initialize the MTD.

Environment
Internal and External

History
Released in NX2

int UF_SIM_mtd_init
(
UF_SIM_engine_p_t engine
)

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object

---

UF_SIM_mtd_reset (view source)

Defined in: uf_sim_commands.h

Overview
Reset the MTD of the simulation engine.

Environment
Internal and External

History
Released in NX2

int UF_SIM_mtd_reset
(
UF_SIM_engine_p_t engine
)

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object

---

UF_SIM_set_axis_allow_reversal (view source)

Defined in: uf_sim_commands.h

Overview
Allow reversal mode for the given NC axis.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_axis_allow_reversal
(
UF_SIM_engine_p_t engine,
const char* axis_name,
logical allow_reversal
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**axis_name**	Input	- name of the NC axis
logical	**allow_reversal**	Input	- switch to set the mode

UF_SIM_set_axis_rotary_dir_mode (view source)

Defined in: uf_sim_commands.h

Overview
Set the direction of rotation mode for the specified NC axis. This setting controls the path (CW or CCW) a rotary axis will follow when it goes to target position. The possible rotation modes are:

UF_SIM_AXIS_ROT_MAGNITUDE_DETERMINES_DIRECTION
The rotary axis behaves like a linear axis where if target position > 0 then it rotate in CCW. If < 0 then it rotates CW. For example if current position is 355 and target position is 370 then it rotates in CCW 15 degrees. If target position is 10 then it rotates in CW 345 degrees.

UF_SIM_AXIS_ROT_ALWAYS_SHORTEST
Moves to the target position in shortest path where possible. This is the default behavior.

UF_SIM_AXIS_ROT_SIGN_DETERMINES_DIRECTION
For positive target position the axis rotates CCW if negative it rotates CW.

UF_SIM_AXIS_ROT_ALWAYS_CLW
Rotates in CW direction only.

UF_SIM_AXIS_ROT_ALWAYS_CCLW
Rotates in CCW direction only.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_axis_rotary_dir_mode
(
UF_SIM_engine_p_t engine,
const char* axis_name,
UF_SIM_axis_rot_dir_type_t rot_mode

)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
const char*	**axis_name**	Input	- name of the axis
UF_SIM_axis_rot_dir_type_t	**rot_mode**	Input	- rot mode to set for axis

---

UF_SIM_set_channel (view source)

Defined in: uf_sim_commands.h

Overview
Set the active channel. All following commands write their data into the chosen channel.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_channel
(
UF_SIM_engine_p_t engine,
int channel_number
)

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
int	**channel_number**	Input	- the number of the active channel

---

UF_SIM_set_coolant (view source)

Defined in: uf_sim_commands.h

Overview
Set the coolant mode in the simulation scenario for multiple purposes e.g., display.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_coolant
(

```
    UF_SIM_engine_p_t engine,
    int value,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
int	**value**	Input	- the value of coolant status 0 is OFF 1 is ON
double	**duration**	Input	- the duration in seconds to complete setting the speed

---

UF_SIM_set_current_zcs_junction (view source)

Defined in: uf_sim_commands.h

Overview
Set the current ZCS junction.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_set_current_zcs_junction
(
    UF_SIM_engine_p_t engine,
    char* junction_name
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
char*	**junction_name**	Input	- the name of the junction

UF_SIM_set_cutting_mode (view source)

Defined in: uf_sim_commands.h

Overview
Set the cutting mode in the simulation scenario.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_set_cutting_mode
(
    UF_SIM_engine_p_t engine,
    UF_SIM_cutting_mode_t mode
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
UF_SIM_cutting_mode_t	**mode**	Input	- the cutting mode

UF_SIM_set_feed (view source)

Defined in: uf_sim_commands.h

Overview

Set the feed rate in the simulation scenario for multiple purposes e.g., display.

Environment

Internal and External

History

Released in NX2

```
int UF_SIM_set_feed
(
    UF_SIM_engine_p_t engine,
    double value,
    UF_SIM_unit_type_t unit,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
double	**value**	Input	- the feed rate value
UF_SIM_unit_type_t	**unit**	Input	- the unit of the feed rate
double	**duration**	Input	- the duration in seconds to complete setting the feed rate

---

UF_SIM_set_immediate_update (view source)

Defined in: uf_sim_commands.h

Overview

Sets whether the processing the events that are put in the input buffer is done right away or they are put in buffer for later processing. By default this mode is control by the simulation engine and should be used very carefully. When this is set to True some of the simulation functions such as simulation speed control and stepping creteria control may not function the

same. This functionality is provided for cases where there is a need to process the events as they are created with no delay of buffering. So, on every call to UF_SIM_update all events will be processed.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_immediate_update
(
UF_SIM_engine_p_t engine,
logical immediate_update
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
logical	**immediate_update**	Input	- whether to process events as they are put in buffer

---

UF_SIM_set_mtd_units (view source)

Defined in: uf_sim_commands.h

Overview
Set the machine tool driver units. Those will be the units to input/output data from S&V commands.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_mtd_units
(
UF_SIM_engine_p_t engine,
UF_SIM_unit_type_t units
)

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
UF_SIM_unit_type_t	**units**	Input	- units to set

UF_SIM_set_parameter (view source)

Defined in: uf_sim_commands.h

Overview

Set the given parameter and store the data into the model.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_parameter
(
UF_SIM_engine_p_t engine,
char* label,
char* text,
UF_SIM_unit_type_t unit,
double duration
)

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
char*	**label**	Input	- the type of the parameter
char*	**text**	Input	- the value of the parameter
UF_SIM_unit_type_t	**unit**	Input	- the unit of the parameter
double	**duration**	Input	- the duration in seconds to complete this action of the parameter setting

---

UF_SIM_set_speed (view source)

Defined in: uf_sim_commands.h

Overview
Set the speed in the simulation scenario for multiple purposes e.g., display.

Environment
Internal and External

History
Released in NX2

int UF_SIM_set_speed
(
UF_SIM_engine_p_t engine,
double value,
UF_SIM_unit_type_t unit,
double duration
)

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
double	**value**	Input	- the speed value

UF_SIM_unit_type_t	**unit**	Input	- the unit of the speed
double	**duration**	Input	- the duration in seconds to complete setting the speed

---

UF_SIM_set_status_send_nc_command_msg (view source)

Defined in: uf_sim_commands.h

Overview
Set the setting for sending NC command message to the feedback processor.
By default it is issued automatically by the system.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_set_status_send_nc_command_msg
(
    UF_SIM_engine_p_t engine,
    logical status
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
logical	**status**	Input	- switch to set the status

UF_SIM_start_of_simulation (view source)

Defined in: uf_sim_commands.h

Overview
Indicate start of simulation.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_start_of_simulation
(
    UF_SIM_engine_p_t engine
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object

---

UF_SIM_step (view source)

Defined in: uf_sim_commands.h

Overview
Create a sub-step event, which indicates the end of an intermediate action. Similar to a step event, all events prior to sub-step event are executed simultaneously.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_step
(
    UF_SIM_engine_p_t engine,
    const char* label
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
const char*	**label**	Input	- name of event

UF_SIM_transform_matrix_acs_to_mtcs (view source)

Defined in: uf_sim_commands.h

Overview
Request the matrix between the two coordinates ACS and MTCS.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_transform_matrix_acs_to_mtcs
(
    UF_SIM_engine_p_t engine,
    double acs_matrix [ 9 ] ,
    double* matrix
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
double	**acs_matrix [9]**	Input	- matrix of a c-system

double*	**matrix**	Output	- matrix between ZCS and MTCS

UF_SIM_transform_offset_acs_to_mtcs (view source)

Defined in: uf_sim_commands.h

Overview
Request the offset between the two coordinates ACS and MTCS.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_transform_offset_acs_to_mtcs
(
    UF_SIM_engine_p_t engine,
    double acs_vector [ 3 ] ,
    double* xval,
    double* yval,
    double* zval
)
```

UF_SIM_engine_p_t	**engine**	Input	- simulation engine object
double	**acs_vector [3]**	Input	- vector of a c-system
double*	**xval**	Output	- offset value in x direction
double*	**yval**	Output	- offset value in y direction
double*	**zval**	Output	- offset value in z direction

UF_SIM_unmount_kim_comp (view source)

Defined in: uf_sim_commands.h

Overview
Un-mount a specified component (source component).

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_unmount_kim_comp
(
```

```
    UF_SIM_engine_p_t engine,
    char* component,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
char*	**component**	Input	- the name of the component to un-mount
double	**duration**	Input	- the time in seconds to complete this un-mount on the machine tool

---

UF_SIM_unmount_tool (view source)

Defined in: uf_sim_commands.h

Overview
Un-mount the tool.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_unmount_tool
(
    UF_SIM_engine_p_t engine,
    char* tool_name,
    double duration
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
char*	**tool_name**	Input	- the name of the tool to un-mount
double	**duration**	Input	- the time in seconds to complete this un-mount on the machine tool

---

UF_SIM_update (view source)

Defined in: uf_sim_commands.h

Overview
Send an update command, which indicates the end of a movement.

Environment
Internal and External

History
Released in NX2

```
int UF_SIM_update
(
    UF_SIM_engine_p_t engine,
    const char* label
)
```

UF_SIM_engine_p_t	**engine**	Input	- the simulation engine object
const char*	**label**	Input	- name of event
