## UF_PATH_create_auxfun (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command AUXFUN/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_auxfun
(
    UF_PATH_id_t path_id,
    int auxfun_code,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| int | **auxfun_code** | Input | Auxfun value |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_circular_motion (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of circular motion (CIRCLE/ ).

### Environment

Internal and External

```
int UF_PATH_create_circular_motion
(
    UF_PATH_id_t path_id,
    UF_PATH_circular_motion_t * circular_motion_data
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Path to perform action on |
| UF_PATH_circular_motion_t * | **circular_motion_data** | Input | Structure defining the circular motion. |

## UF_PATH_create_clamp (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command CLAMP/. The appended text is optional. If "text" is NULL then no appended test appears.

### Environment

Internal and External

```
int UF_PATH_create_clamp
(
    UF_PATH_id_t path_id,
    UF_PATH_clamp_status_t clamp_status,
    UF_PATH_axis_t axis_type,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_clamp_status_t | **clamp_status** | Input | The clamp status. |
| UF_PATH_axis_t | **axis_type** | Input | The axis type. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_contact_circular_motion (view source)

**Defined in: uf_path.h**

### Overview

This function creates a contact circular motion.

### Environment

Internal and External

```
int UF_PATH_create_contact_circular_motion
(
    UF_PATH_id_t path_id,
    UF_PATH_circular_motion_t * motion_data,
    UF_PATH_contact_data_t * contact_data
)
```

| | |
|---|---|
| UF_PATH_id_t | **path_id** |
| UF_PATH_circular_motion_t * | **motion_data** |
| UF_PATH_contact_data_t * | **contact_data** |

## UF_PATH_create_contact_linear_motion (view source)

**Defined in: uf_path.h**

### Overview

This function creates a contact linear motion.

### Environment

Internal and External

```
int UF_PATH_create_contact_linear_motion
(
    UF_PATH_id_t path_id,
    UF_PATH_linear_motion_t * linear_motion_data,
    UF_PATH_contact_data_t * contact_data
)
```

| | |
|---|---|
| UF_PATH_id_t | **path_id** |
| UF_PATH_linear_motion_t * | **linear_motion_data** |
| UF_PATH_contact_data_t * | **contact_data** |

---

## UF_PATH_create_coolant_off (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command COOLNT/OFF. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_coolant_off
(
    UF_PATH_id_t paht_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **paht_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_coolant_on (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command COOLNT/.

### Environment

Internal and External

```
int UF_PATH_create_coolant_on
(
    UF_PATH_id_t path_id,
    UF_PATH_coolant_type_t coolant_type,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_coolant_type_t | **coolant_type** | Input | The coolant type. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_cut_wire (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command UNLOAD/WIRE. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_cut_wire
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_cutcom (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command CUTCOM/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_cutcom
(
    UF_PATH_id_t path_id,
    UF_PATH_cutcom_t * cutcom_data,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_cutcom_t * | **cutcom_data** | Input | Structure specifying the cutcom data. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_dwell (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command DELAY/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_dwell
(
    UF_PATH_id_t path_id,
    double dwell_value,
    UF_PATH_dwell_unit_t dwell_unit,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| double | **dwell_value** | Input | Amount of delay |
| UF_PATH_dwell_unit_t | **dwell_unit** | Input | The units to use for the delay. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_fedrat (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of the post command FEDRAT/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_fedrat
(
    UF_PATH_id_t path_id,
    double fedrat_value,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| double | **fedrat_value** | Input | Feed rate value |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_flush (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command FLUSH/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_flush
(
    UF_PATH_id_t path_id,
    UF_PATH_flush_t * flush_data,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_flush_t * | **flush_data** | Input | The flush data. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_flush_tank (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command FLUSH/IN. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_flush_tank
(
    UF_PATH_id_t path_id,
    UF_PATH_tank_type_t tank_type,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_tank_type_t | **tank_type** | Input | Specifies the tank type. |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_helical_motion (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of helical motion (CIRCLE/.....,TIMES ).

### Environment

Internal and External

```
int UF_PATH_create_helical_motion
(
    UF_PATH_id_t path_id,
    UF_PATH_helical_motion_t * helical_motion_data
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Path to perform action on |
| UF_PATH_helical_motion_t * | **helical_motion_data** | Input | Structure defining the helical motion. |

---

## UF_PATH_create_level_marker (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of level marker.

### Environment

Internal and External

```
int UF_PATH_create_level_marker
(
    UF_PATH_id_t path_id,
    UF_PATH_level_marker_t * level_marker_data
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Path to perform action on |
| UF_PATH_level_marker_t * | **level_marker_data** | Input | Structure specifying the level marker data |

## UF_PATH_create_linear_motion (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of linear motion( GOTO/ ).

### Environment

Internal and External

```
int UF_PATH_create_linear_motion
(
    UF_PATH_id_t path_id,
    UF_PATH_linear_motion_t * linear_motion_data
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Path to perform action on |
| UF_PATH_linear_motion_t * | **linear_motion_data** | Input | Structure specifying the linear motion |

## UF_PATH_create_op_skip (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post comand OPSKIP/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_op_skip
(
    UF_PATH_id_t path_id,
    UF_PATH_op_skip_t skip_option,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |

| | | | |
|---|---|---|---|
| UF_PATH_op_skip_t | **skip_option** | Input | The skip option. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_op_stop (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command OPSTOP. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_op_stop
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_opmessage (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of the post command DISPLY/.

### Environment

Internal and External

```
int UF_PATH_create_opmessage
(
    UF_PATH_id_t path_id,
    const char* text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char* | **text** | Input | The text to output to the operator. |

## UF_PATH_create_origin (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of the post command ORIGIN/. The appended text is optional. If 'text' is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_origin
(
    UF_PATH_id_t path_id,
    const double origin_coordinates [ 3 ] ,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const double | **origin_coordinates [ 3 ]** | Input | The origin x/y/z |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_power (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command POWER/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_power
(
    UF_PATH_id_t path_id,
    double power_value,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| double | **power_value** | Input | Value of the power register |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_pprint (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command PPRINT/.

### Environment

Internal and External

```
int UF_PATH_create_pprint
(
    UF_PATH_id_t path_id,
    const char* text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char* | **text** | Input | The text to output on the PPRINT command |

---

## UF_PATH_create_prefun (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command PREFUN/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_prefun
(
    UF_PATH_id_t path_id,
    int prefun_code,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| int | **prefun_code** | Input | Prefun value |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_rotate (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command ROTATE/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_rotate
(
    UF_PATH_id_t path_id,
    UF_PATH_rotate_t * rotate_data,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path ID for this event |
| UF_PATH_rotate_t * | **rotate_data** | Input | The data specifying the rotation. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_select_head (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SELECT/HEAD. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_select_head
(
    UF_PATH_id_t path_id,
    UF_PATH_head_type_t head_type,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_head_type_t | **head_type** | Input | The head type in the turret of the lathe. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_seqno (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SEQNO/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_seqno
(
    UF_PATH_id_t path_id,
    UF_PATH_seqno_t seq_type,
    int seq_number,
    int seq_incr,
    int seq_freq,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_seqno_t | **seq_type** | Input | The sequence type |
| int | **seq_number** | Input | The sequence number, only valid if seq_type is UF_PATH_SEQNO_N |
| int | **seq_incr** | Input | Only valid if seq_type is UF_PATH_SEQNO_N |
| int | **seq_freq** | Input | Only valid if seq_type is UF_PATH_SEQNO_N |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_set_mode (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SET/. The appended text is optional. If 'text' is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_set_mode
(
    UF_PATH_id_t path_id,
    UF_PATH_output_mode_t ouput_mode,
    UF_PATH_feedrate_mode_t feedrate_mode,
    UF_PATH_arc_mode_t arc_mode,
    UF_PATH_parallel_mode_t parallel_mode,
    UF_PATH_machine_mode_t machine_mode,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_output_mode_t | **ouput_mode** | Input | The output mode for this tool path. |
| UF_PATH_feedrate_mode_t | **feedrate_mode** | Input | The feedrate mode for this tool path. |
| UF_PATH_arc_mode_t | **arc_mode** | Input | The arc mode for this tool path. |
| UF_PATH_parallel_mode_t | **parallel_mode** | Input | The parallel mode for this tool path. |
| UF_PATH_machine_mode_t | **machine_mode** | Input | The machine mode for this tool path. |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_spindle_off (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SPINDL/OFF. The appended text is optional. If "text" is NULL then bo appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_spindle_off
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_spindle_on (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SPINDL/. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_spindle_on
(
    UF_PATH_id_t path_id,
    UF_PATH_spindle_on_t * spindle_on_data,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_spindle_on_t * | **spindle_on_data** | Input | The spindle data for the command. |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_spindle_reverse (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SPINDL/REVERSE.
The appended text is optional.
If "text" is NULL then bo appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_spindle_reverse
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

---

## UF_PATH_create_stop (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command STOP. The appended text is optional. If the "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_stop
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_text (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of a post command text string.

### Environment

Internal and External

```
int UF_PATH_create_text
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command text string |

## UF_PATH_create_thread_wire (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command LOAD/WIRE. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_thread_wire
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_tool_change (view source)

**Defined in: uf_path.h**

### Overview

This functions allows the creation of post command LOAD/TOOL(for mill) or TURRET/ (for lathe). The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_tool_change
(
    UF_PATH_id_t path_id,
    UF_PATH_tool_change_t * tool_change_data,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_tool_change_t * | **tool_change_data** | Input | Data specifying the tool change to make. |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_tool_length_comp (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SET/ADJUST. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_tool_length_comp
(
    UF_PATH_id_t path_id,
    int tool_comp_register,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| int | **tool_comp_register** | Input | Tool length compensation register value |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_tool_preselect (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SELECT/TOOL. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_tool_preselect
(
    UF_PATH_id_t path_id,
    int tool_number,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| int | **tool_number** | Input | Tool number value |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_create_tracking_point_change (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command TRACKING POINT CHANGE.
The appended text is optional.
If "text" is NULL then bo appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_tracking_point_change
(
    UF_PATH_id_t path_id,
    UF_PATH_tracking_point_change_t * tool_change_data
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| UF_PATH_tracking_point_change_t * | **tool_change_data** | Input | Data specifying the tracking point change to make. |

# UF_PATH_create_wire_angle (view source)

**Defined in: uf_path.h**

## Overview

This function allows the creation of post command STAN/. The appended text is optional. If "text" is NULL then no appended text appears.

## Environment

Internal and External

```
int UF_PATH_create_wire_angle
(
    UF_PATH_id_t path_id,
    double slope_value,
    double angle_value,
    logical angle_flag,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| double | **slope_value** | Input | Slope of the wire |
| double | **angle_value** | Input | Angle of the wire |
| logical | **angle_flag** | Input | Flag to indicate if angle is specified |
| const char * | **text** | Input | The post command appended text |

# UF_PATH_create_wire_cutcom (view source)

**Defined in: uf_path.h**

## Overview

This function allows the creation of post command CUTCOM/. The appended text is optional. If "text" is NULL then no appended text appears.

## Environment

Internal and External

```
int UF_PATH_create_wire_cutcom
(
    UF_PATH_id_t path_id,
    UF_PATH_cutcom_mode_t cutcom_mode,
    int adjust_register,
    logical cutcom_off_flag,
    logical adjust_flag,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |

| | | | |
|---|---|---|---|
| UF_PATH_cutcom_mode_t | **cutcom_mode** | Input | The mode for the CUTCOM command. |
| int | **adjust_register** | Input | Adjust register value |
| logical | **cutcom_off_flag** | Input | Flag indicating if cutcom off is to be output |
| logical | **adjust_flag** | Input | Flag to indicate if the adjust register is specified |
| const char * | **text** | Input | The post comand appended text |

## UF_PATH_create_wire_guides (view source)

**Defined in: uf_path.h**

### Overview

This function allows the creation of post command SET/UPPER. The appended text is optional. If "text" is NULL then no appended text appears.

### Environment

Internal and External

```
int UF_PATH_create_wire_guides
(
    UF_PATH_id_t path_id,
    const char * text
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | The path id for this event |
| const char * | **text** | Input | The post command appended text |

## UF_PATH_end_tool_path (view source)

**Defined in: uf_path.h**

### Overview

This function signifies the end of the tool path. The "oper_id" is no longer valid after this function executes.

### Environment

Internal and External

```
int UF_PATH_end_tool_path
(
    UF_PATH_id_t path_id
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Identifier of the path to end |

## UF_PATH_init_tool_path (view source)

**Defined in: uf_path.h**

### Overview

This function initializes the tool path header.

### Environment

Internal and External

```
int UF_PATH_init_tool_path
(
    UF_PATH_id_t path_id
)
```

| | | | |
|---|---|---|---|
| UF_PATH_id_t | **path_id** | Input | Identifier passed from UF function |