UF_SETUP_ask_geom_null (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. Each of these views has a special Group known as the NULL Group. In the Geometry view the NULL Group contains those operations for which no geometry is currently assigned. This function returns the NULL Group of the Geomtry view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_geom_null
(
    tag_t setup_tag,
    tag_t * geom_null
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**geom_null**	Output	- the NULL Group of the geometry view

---

UF_SETUP_ask_geom_root (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. This function returns the root of the Geometry view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_geom_root
(
    tag_t setup_tag,
    tag_t * geom_group
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**geom_group**	Output	- the root of the geometry view of the specified setup object

UF_SETUP_ask_mct_null (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. Each of these views has a special Group known as the NULL Group. In the Machine Tool view the NULL Group contains those operations for which no cutter is currently assigned. However, the operations in this Group are assigned to the Machine Tool represented by the root of this view. This function returns the NULL Group of the Machine Tool view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_mct_null
(
    tag_t setup_tag,
    tag_t * mct_null
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**mct_null**	Output	- the NULL Group of the machine tool view

UF_SETUP_ask_mct_root (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. This function returns the root of the Machine Tool view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_mct_root
(
    tag_t setup_tag,
    tag_t * mct_group
)
```

tag_t	**setup_tag**	Input	- the setup object to query

tag_t *	**mct_group**	Output	- the root of the machine tool view of the specified setup object

---

UF_SETUP_ask_mthd_null (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. Each of these views has a special Group known as the NULL Group. The NULL Group of the Machining Method has no specific use; it may be used anyway the user wishes. This function returns the NULL Group of the Machining Method view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_mthd_null
(
    tag_t setup_tag,
    tag_t * mthd_null
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**mthd_null**	Output	- the NULL Group of the machining method view

---

UF_SETUP_ask_mthd_root (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. This function returns the root of the Machining Method view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_mthd_root
(
    tag_t setup_tag,
    tag_t * mthd_group
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**mthd_group**	Output	- the root of the machining method view of the specified setup object

UF_SETUP_ask_program_null (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. Each of these views has a special Group known as the NULL Group. In the Program view the NULL Group contains those operations for which no program should currently be generated. If one asks to generate all programs for the Program root, operations in the NULL Group will be ignored. This function returns the NULL Group of the Program view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_program_null
(
    tag_t setup_tag,
    tag_t * program_null
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**program_null**	Output	- the NULL Group of the program view

UF_SETUP_ask_program_root (view source)

Defined in: uf_setup.h

Overview

Each UF_SETUP has four views: Program, Geometry, Machine Tool and Machining Method. This function returns the root of the Program view.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_program_root
(
    tag_t setup_tag,
    tag_t * program_group
)
```

tag_t	**setup_tag**	Input	- the setup object to query
tag_t *	**program_group**	Output	- the root of the program view of the specified setup object

---

UF_SETUP_ask_setup (view source)

Defined in: uf_setup.h

Overview

This function returns the current UF_SETUP object. This object is the setup in the current part.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_ask_setup
(
    tag_t * setup_tag
)
```

tag_t *	**setup_tag**	Output	- the current setup object

UF_SETUP_create (view source)

Defined in: uf_setup.h

Overview

This function creates a UF_SETUP object in the part specified by 'part_tag'. It uses the part 'template_name' as its template. The setup, views, groups and operations of 'template_name' become the setup of 'part_tag'. No geometry is loaded and no attempt is made to map the geometry in 'template_name' to the geometry in 'part_tag'.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_create
(
        tag_t part_tag,
        const char * template_name
    )
```

tag_t	**part_tag**	Input	- see above
const char *	**template_name**	Input	- see above

UF_SETUP_delete_setup (view source)

Defined in: uf_setup.h

Overview

This function deletes the current UF_SETUP object. This object is the setup in the current part.

Environment

Internal and External

History

Originally released in V18.0

```
    int UF_SETUP_delete_setup
    (
        void
    )
```

UF_SETUP_generate_clsf (view source)

Defined in: uf_setup.h

Overview

This function generates the CLSF for all the operations contained in the specified group of the specified setup. It will use the instructions (i.e., the Definition File and Event Handler) specified by 'clsf_name'. It will write the generated CLSF to a file with name 'output_filename'. If 'output_filename' exists it will be erased before writing new data to it.

See the Note in the File Description regarding the relationship between the 'clsf_name' and the CAM resource files.

Environment

Internal and External

History

Originally released in V16.0

```
    int UF_SETUP_generate_clsf
    (
    tag_t setup,
    tag_t group,
    const char * clsf_name,
    const char * output_filename,
    UF_SETUP_output_units_t output_units
)
```

tag_t	**setup**	Input	- the setup object to process
tag_t	**group**	Input	- the program group to process
const char *	**clsf_name**	Input	- the name (as found in a resource file) of the CLSF instructions to use. This name specifies the Definition File and Event Handler file to use
const char *	**output_filename**	Input	- the name of the file to write the output to
UF_SETUP_output_units_t	**output_units**	Input	- the output units for the Clsf processor to use

---

UF_SETUP_generate_program (view source)

Defined in: uf_setup.h

Overview

This function generates the NC program for all the operations contained in the specified group of the specified setup. It will use the Post (i.e., the Definition File and Event Handler) specified by 'post_name'. It will write the generated NC Program to a file with name 'output_filename'. If 'output_filename' exists the output will be appended to the existing file. If the behavior wanted is to overwrite the exsiting file, then the user has to delete the file before calling this function.

See the Note in the File Description regarding the relationship between the 'post_name' and the CAM resource files.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SETUP_generate_program
(
    tag_t setup,
    tag_t group,
    const char * post_name,
    const char * output_filename,
    UF_SETUP_output_units_t output_units
)
```

tag_t	**setup**	Input	- the setup object to process

tag_t	**group**	Input	- the program group to process
const char *	**post_name**	Input	- the name (as found in a resource file) of the post instructions to use. This name specifies the Definition File and Event Handler file to use
const char *	**output_filename**	Input	- the name of the file to write the output to
UF_SETUP_output_units_t	**output_units**	Input	- the output units for the Post to use

UF_SHOPDOC_generate_doc (view source)

Defined in: uf_setup.h

Overview

This function generates Shop Floor Documentation based upon the Template file and Event Handler file specified by 'doc_name'. It will write the data generated by the Template file and Event Handler file to a file with name 'output_filename'. If 'output_filename' exists it will be erased before writing new data to it.

See the Note in the File Description regarding the relationship between the 'doc_name' and the CAM resource files.

Environment

Internal and External

History

Originally released in V16.0

```
int UF_SHOPDOC_generate_doc
(
    const char * doc_name,
    const char * output_filename,
    UF_SETUP_output_units_t output_units
)
```

const char *	**doc_name**	Input	- the name (as found in a resource file) of the documentation instructions to use. This name specifies the name of the Template file and Event Handler file to use.
const char *	**output_filename**	Input	- the name of the file to write the output to
UF_SETUP_output_units_t	**output_units**	Input	- the output units for the Shop documentation to use