UF_ABORT_ask_flag_status (view source)

Defined in: uf_abort.h

Overview

This routine is used to check whether the user has signaled an abort request
by pressing the key sequence CTRL+SHIFT+L or selecting OK in the Work
In Progress dialog if NX has not had a chance to act on the signal.

But if NX has already processed the abort request by the time this function
is called, then the return code is FALSE as the abort signal is already reset.

Return

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

Environment

Internal and External

History

Introduced in V16.0.3

Required License(s)

gateway

```
int UF_ABORT_ask_flag_status
(
    logical * flag
)
```

logical *	**flag**	Output	when set to TRUE, indicates an abort signal

UF_ABORT_clear_abort (view source)

Defined in: uf_abort.h

Overview

This routine is used to clear any pending abort requests. It should be
called if the abort request is handled.

Return

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

Environment

Internal and External

History

Introduced in V16.0.3

Required License(s)

gateway

```
int UF_ABORT_clear_abort
(
    void
)
```

UF_ABORT_disable_abort (view source)

Defined in: uf_abort.h

Overview

This routine is used to disable the abort handler. It must be re-enabled with the function, UF_ABORT_enable_abort.

Return

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

Environment

Internal and External

History

Introduced in V18.0

Required License(s)

gateway

```
int UF_ABORT_disable_abort
(
    void
)
```

UF_ABORT_enable_abort (view source)

Defined in: uf_abort.h

Overview

This routine is used to enable the abort handler. It is used in conjunction with UF_ABORT_disable_abort.

Return

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

Environment

Internal and External

History

Introduced in V18.0

Required License(s)

gateway

```
int UF_ABORT_enable_abort
(
    void
)
```