

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

Defined in: `uf.h`

Also known as:

- `UF_callback_reason_e_t`

Overview

These are the reasons against which call backs can be registered.

The registered callbacks will be called whenever the specific reason occurs in interactive NX.

For all the reasons, except for `UF_close_part_reason`, the callbacks will be called towards the end of that particular operation. For example, the callbacks registered against `UF_create_part_reason` will be called when a new part is created, after almost all the process of creating this new part is completed.

The `UF_change_work_part_reason` call back will be called after a work part change has occurred. It should be noted that this callback may be called at times where the old work part (which is passed as the part tag data to the call back routine) may not necessarily be different from the new work part. For example, this call back is also called if the display part has changed but the new work part is still the same tag and now represents a component in a different display part. This call back may be called more than once as part of a single operation.

An attempt to query the work or display part in a `UF_save_part_reason_callback` may return unexpected results.

The callbacks registered against `UF_close_part_reason` will be called at the beginning of part close.

The callbacks registered against `UF_modified_part_reason` will be called when the part is modified for the first time after opening it or after a save operation.

These callbacks give a handle to the user to do some extra processing, in addition to the processing that NX does, during that particular operation.

In NX Manager mode, when a new part is being created it first loads the template part (seed part), changes the template part to be the work part and then renames the work part to the new part name given. So, the following callbacks are being called:

`UF_modified_part_reason`, `UF_open_part_reason`, `UF_change_work_part_reason`, `UF_rename_part_reason` and `UF_create_part_reason`.

Data Members

`UF_create_part_reason`

Callback when a new part is created.

`UF_open_part_reason`

Callback when an existing part is opened.

`UF_save_part_reason`

Callback when a part is saved.

`UF_save_as_part_reason`

Callback when a part is saved-as.

UF_close_part_reason

Callback when a part is closed.

UF_modified_part_reason

Callback when a part is modified.

UF_rename_part_reason

Callback when a part is renamed.

UF_change_work_part_reason

Callback when there is a new work part.

The part returned is of the old work part.

Use UF_ASSEM_ask_work_part to get new work part.

UF_post_save_as_part_reason

Callback reason for adding parts for

Save As at the end of in progress Save As operation

of selected part/parts, it's like nested Save As.

Only issued in managed mode.

UF_max_reason

Do not use - not a valid reason

UF_codeset_e ([view source](#))

Defined in: `uf.h`

Also known as:

- `UF_codeset_t`

Overview

Codesets supported by NX. This is returned by the function `UF_ask_codeset`.

Data Members**UF_CODESET_NOT_INITIALIZED=-2**

This is an error condition.

UF_CODESET_NOT_SUPPORT=-1

This is an error condition.

UF_CODESET_ASCII

Only ASCII characters are supported.

If NX does not recognize a codeset, then it will revert to `UF_CODESET_ASCII`.

UF_CODESET_ISO8859_1

Western European standard

UF_CODESET_EUC

Japanese EUC codeset

UF_CODESET_SJIS

Japanese SJIS codeset

UF_CODESET_ISO8859_4

Finnish, Latvian, Norwegian, Swedish
and other European languages.

UF_CODESET_ISO8859_5

Russian, Ukrainian, Bulgarian and other
Eastern European languages.

UF_CODESET_ISO8859_8

Hebrew codeset

UF_CODESET_BIG5

Traditional Chinese for Taiwan

UF_CODESET_NT_866

No longer used, placeholder only

UF_CODESET_ISO8859_2

Czech, Polish, Slovak and other
Eastern European languages.

UF_CODESET_NT_852

No longer used, placeholder only

UF_CODESET_NT_862

No longer used, placeholder only

UF_CODESET_KSC5601

Korean codeset

UF_CODESET_GB2312

Simplified Chinese for Mainland China

UF_CODESET_NT_1251

Windows NT Russian codeset

UF_CODESET_NT_1255

Windows NT Hebrew codeset

UF_CODESET_NT_1250

Windows NT Eastern European codeset

UF_CODESET_NT_1252

Windows NT Western European and US
codeset

UF_CODESET_ISO8859_15

Western European codeset including
EURO

UF_CODESET_NT_1253

Windows NT Greek codeset

UF_CODESET_ISO8859_7

Greek Codeset

UF_CODESET_UTF8

UTF-8 Codeset

UF_CODESET_NT_1254

Windows NT Turkish codeset

UF_CODESET_NT_1257

Windows NT Baltic codeset

UF_CODESET_NT_1256

Windows NT Arabic codeset

UF_CODESET_ISO8859_6

Arabic Codeset

UF_CODESET_ISO8859_11

Thai Unix and Windows (CP 874)

UF_CODESET_NT_1258

Vietnamese codeset

UF_TAG_event_e ([view source](#))Defined in: `uf.h`

Also known as:

- `UF_TAG_event_t`

Overview

This enum identifies the type of an event that occurred on a tag.

Data Members**UF_TAG_EVENT_NORMAL_CREATE**

The tag was created.

UF_TAG_EVENT_UNDO_OVER_CREATE

An undo occurred over the tag's creation and so is now invalid.

UF_TAG_EVENT_NORMAL_DELETE

The tag was deleted.

UF_TAG_EVENT_UNDO_OVER_DELETE

An undo occurred over the tag's deletion and so has come back to life.

UF_TAG_EVENT_UNDO_DELETE_EXPIRED

NX will now never undo over the deletion so it will never come back to life.

UF_TAG_EVENT_UNDO_CREATE_EXPIRED

NX will now never undo over the tags creation.
