zy_err

NAME

Error Dequeue Abstraction Library

LIBRARY

zy_err (-lzy_err)

SYNOPSIS

```
#include <zy_err.h>

typedef struct zy_err_bx_s zy_err_bx_t;
typedef struct zy_err_s zy_err_t;

int zy_err_construct(zy_err_t **err, const zyalloc_t *alloc);
void zy_err_destruct(zy_err_t **err);
void zy_err_clear(zy_err_t *err);
int zy_err_push_first(zy_err_t *err, int64_t code, const char *file,
                      size_t line, const char *function,
                      const void *opaque, size_t opaque_size);
int zy_err_push_last(zy_err_t *err, int64_t code, const char *file,
                     size_t line, const char *function,
                     const void *opaque, size_t opaque_size);
void zy_err_discard_first(zy_err_t *err);
void zy_err_discard_last(zy_err_t *err);
zy_err_bx_t *zy_err_peek_first(const zy_err_t *err);
zy_err_bx_t *zy_err_peek_last(const zy_err_t *err);
size_t zy_err_size(const zy_err_t *err);
bool zy_err_is_empty(const zy_err_t *err);

int64_t zy_err_bx_code(const zy_err_bx_t *bx);
const char *zy_err_bx_file(const zy_err_bx_t *bx);
size_t zy_err_bx_line(const zy_err_bx_t *bx);
const char *zy_err_bx_function(const zy_err_bx_t *bx);
const void *zy_err_bx_opaque(const zy_err_bx_t *bx, size_t *size);
```

DESCRIPTION

zy_err_construct()

`zy_err_construct` allocates a `zy_err_t` data structure using `alloc` and stores the result in `*err`. All function arguments must be non-null.

zy_err_destruct()

`zy_err_destruct` deallocates a `zy_err_t` data structure and sets `*err` to `nullptr`. Note that `err` must be non-null.

zy_err_clear()

`zy_err_clear` deallocates and unlinks all `zy_err_bx_t` data structures stored in `err`. Note that `err` must be non-null.

zy_err_push_*()

`zy_err_push_first` and `zy_err_push_last` allocate a `zy_err_bx_t` and store `code`, `file`, `line`, `function`,`opaque`, and `opaque_size`.

The resulting data structure is stored at the *front* and *back* of `err` for `zy_err_push_first` and `zy_err_push_last`, respectively.

Note that `err`, `file`, and `function` must be non-null and `line` must be non-zero. `opaque` may be set to `nullptr` and `opaque_size` to zero in order to indicate that there is no auxiliary data; however, if `opaque` is non-null, then `opaque_size` must be non-zero.

zy_err_discard_first()

`zy_err_discard_first` deallocates and unlinks the *front-most* `zy_err_bx_t` data structure from `err`. Note that `err` must be non-null.

zy_err_discard_last()

`zy_err_discard_last` deallocates and unlinks the *back-most* `zy_err_bx_t` data structure from `err`. Note that `err` must be non-null.

zy_err_peek_first()

`zy_err_peek_first` retrieves the *front-most* `zy_err_bx_t` data structure from `err` if it exists. All function arguments must be non-null.

zy_err_peek_last()

`zy_err_peek_last` retrieves the *back-most* `zy_err_bx_t` data structure from `err` if it exists. All function arguments must be non-null.

zy_err_size()

`zy_err_size` returns the number of elements stored in `err`. Note that `err` must be non-null.

zy_err_is_empty()

`zy_err_is_empty` returns a `true` if and only if there are no elements stored in `err`. Note that `err` must be non-null.

zy_err_bx_code()

`zy_err_bx_code` returns the *error code* associated with `bx`. Note that `bx` must be non-null.

zy_err_bx_file()

`zy_err_bx_file` returns the *file* associated with `bx`. Note that `bx` must be non-null.

zy_err_bx_line()

`zy_err_bx_line` returns the *line number* associated with `bx`. Note that `bx` must be non-null.

zy_err_bx_function()

`zy_err_bx_function` returns the *function name* associated with `bx`. Note that `bx` must be non-null.

zy_err_bx_opaque()

`zy_err_bx_opaque` returns the memory address of the opaque data associated with `bx` and stores its size in `*size`. Note that `bx` must be non-null. `size` may be `nullptr` to indicate that the size is known ahead of time and is unneeded.

RETURN VALUE

On success, `zy_err_construct`, `zy_err_push_first`, `zy_err_push_last` return `0`. Otherwise, an error code is returned.

ERRORS

`zy_err_construct`, `zy_err_push_first`, and `zy_err_push_last` can fail with the following error.

ZY_ENOMEM Out of memory.

NOTES

It is undefined behavior to violate any preconditions of these functions (e.g. passing `nullptr` to a function argument that is specified as non-null).