Message queues

# Create a queue.

feabhOS\_error

feabhOS\_queue\_create(feabhOS\_QUEUE \* const queue\_handle,

size\_bytes\_t elem\_size,

num\_elements\_t queue\_size);

## Behaviour

Queues are marshalling - that is, data is copied into the queue on insertion.

Queues are fixed size - both the number of elements and the size of each element.

## Parameters

## queue\_handle A pointer to a feabhOS\_QUEUE object

## elem\_size The size of each queue element (in bytes)

## queue\_size The number of elements in the queue

## Return values

ERROR\_OK Success. Queue handle will be non-NULL

ERROR\_OUT\_OF\_MEMORY Could not allocate memory for the queue

ERROR\_PARAM1 elem\_size == 0

ERROR\_PARAM2 queue\_size == 0

## Additional info

The maximum number of queues that can be created is defined in feabhOS\_port\_defs.h.

Return values may be converted to C-style strings with feabhOS\_error\_as\_string()

# Insert into a queue.

feabhOS\_error

feabhOS\_queue\_post(feabhOS\_QUEUE \* const queue\_handle,

void \* const in,

duration\_mSec\_t timeout);

## Behaviour

Data is copied into the queue. If the queue is full the caller will be suspended until there is space.

## Parameters

queue\_handle A pointer to a feabhOS\_QUEUE object

in A pointer to the data.

timeout Specify the maximum time the caller will block

May be set to

NO\_WAIT non-blocking wait

WAIT\_FOREVER block forever (only return on insertion)

## Return values

ERROR\_OK Success. Data is inserted into the queue.

ERROR\_QUEUE\_FULL Timeout duration expired. No data inserted.

ERROR\_INVALID\_HANDLE queue\_handle == NULL

ERROR\_PARAM1 in == NULL

## Additional info

Return values may be converted to C-style strings with feabhOS\_error\_as\_string()

# Retrieve from a queue.

feabhOS\_error

feabhOS\_queue\_get(feabhOS\_QUEUE \* const queue\_handle,

void \* const in\_out,

duration\_mSec\_t timeout);

## Behaviour

Data is copied into the in-out parameter. If the queue is empty the caller will be suspended until there is data.

## Parameters

queue\_handle A pointer to a feabhOS\_QUEUE object

in\_out A pointer to an object that will receive the data.

timeout Specify the maximum time the caller will block

May be set to

NO\_WAIT non-blocking wait

WAIT\_FOREVER block forever (only return on insertion)

## Return values

ERROR\_OK Success. Data will be pointed to by in\_out.

ERROR\_QUEUE\_EMPTY Timeout duration expired. No data retrieved.

ERROR\_INVALID\_HANDLE queue\_handle == NULL

ERROR\_PARAM1 in\_out == NULL

## Additional info

Return values may be converted to C-style strings with feabhOS\_error\_as\_string()

# Get queue size

num\_elements\_t

feabhOS\_queue\_size(feabhOS\_QUEUE \* const queue\_handle);

## Behaviour

The queue size is the number of elements currently in the queue, NOT the maximum number of elements it can hold (its capacity)

## Parameters

queue\_handle A pointer to a feabhOS\_QUEUE object

## Return values

num\_elements The number of elements in the queue.

num\_elements = 0 if queue\_handle == NULL

## Additional info

num\_elements\_t is defined in "feabhOS\_stdint.h

# Destroy a queue

feabhOS\_error

feabhOS\_queue\_destroy(feabhOS\_QUEUE \* const queue\_handle);

## Behaviour

Destroy the queue object and deallocate any memory for its management structure.

## Parameters

queue\_handle A pointer to a feabhOS\_QUEUE object

## Return values

ERROR\_OK Success. Data will be pointed to by in\_out.

ERROR\_INVALID\_HANDLE queue\_handle == NULL

## Additional info