

## 1. Changes in TR24731-2 Since Registration

This document is intended to track the changes applied to TR24731-2 since the registration ballot.

Changes are listed by section number, and referenced sections are to the *current* draft, and not the registration document, since sections may move.

### 1.1 Changes in Front Matter

The Introduction has been updated to include paragraph 6, warning that the part 1 functions suffer if the bounds value is incorrect, while the allocation functions will not suffer from this problem. However, they do present a denial of service attack not present with the bounds check functions if dynamic memory is exhausted.

### 1.2 Changes in Section 1

None.

### 1.3 Changes in Section 2

Normative references have been updated to show that ISO/IEC 23360 is now an IS, and no longer a DIS.

A normative reference to TC2 has been added.

Several references that do not belong have been removed:

ISO 31-11 Quantities and units - Part 11: Mathematical signs and symbols for use in the physical sciences and technology.

ISO 4217 Codes for the representation of currencies and funds.

ISO 8601 Data elements and interchange formats - Information interchange - Representation of dates and times.

IEC 60559 Binary floating-point arithmetic for microprocessor systems.

### 1.4 Changes in Section 3

Removed mention of mathematical symbols (there aren't any).

### 1.5 Changes in Section 4

None.

The implementation defines `__STDC_ALLOC_LIB_`, the application must define `__STDC_WANT_LIB_EXT2_`. Should these names be aligned (either `__STD_LIB_EXT2_` or `__STDC_WANT_ALLOC_LIB_`)?

### 1.6 Changes in Section 5

1.6.1 Changes in Section 5.2.1 The text has been updated to match the current POSIX draft.

The `open_wmemstream` function is discussed here, despite the fact that it is a `<wchar.h>` function and this is `<stdio.h>`. Is this OK?

1.6.2 Changes in Section 5.2.2 The text has been updated to match the current POSIX draft.

1.6.3 Changes in Section 5.2.3 The `asprintf` family of functions is not in POSIX, but is in the LSB. The `vasprintf` function has been moved to the correct place alongside `asprintf`. The `fscanf` function has been updated to match POSIX, and includes as yet unapproved changes from the most recent ballot (by the time of the Kona meeting, these changes will have been approved).

1.6.4 Changes in Section 5.3.1 The text has been updated to match the current POSIX draft. A "Returns" section has been added to the `strdup` function.

*1.6.5 Changes in Section 5.4.1* The text has been updated to match the current POSIX draft, and to match the narrow equivalent functions.

The wide character versions of `asprintf` have been added. These are pure invention, but give the required symmetry (see action item from London).

On the subject of invention, the `getwdelim` and `getwline` functions are also invention.

## 2. *Other Comments*

There was talk at one point of adding some additional functions to help manage the allocator (see N1085), in particular:

- `sizeof_alloc`
- `request_malloc`
- `resize_alloc`
- `negotiate_alloc`
- `memalign` (based on `posix_memalign`, **not in N1085**).

These have been discussed at earlier meetings, and it was decided to omit them from this TR. All except `sizeof_alloc` and `memalign` are of debatable value, and are not found in most C library implementations. We agreed in Mt Tremblant that N1085 deserved further attention. In Portland we pushed it to WG21, with the addition of `memalign`, with the "request" and "negotiate" functions removed.

Should these be added to TR 24731-2?