

# Debug printing library

This library provides a lightweight printf function that can be enabled or disabled via configuration defines. Code can be declared to be within a “debug unit” (usually a library or application source base) and prints can be enabled/disabled per debug unit.

---

## Features

- Low memory usage
- Ability to enable or disable printing via compile options
- Ability to enable or disable printing for sets of source files

## Software version and dependencies

This document pertains to version 2.1.0 of this library. It is known to work on version 14.2.3 of the xTIMEcomposer tools suite, it may work on other versions.

The library does not have any dependencies (i.e. it does not rely on any other libraries).

## 1 API

To use this module, include `lib_logging` in the application's `USED_MODULES` list and include the `debug_print.h` header file.

### 1.1 Debug printf

Function	<code>debug_printf</code>
Description	<p>A limited functionality version of <code>printf</code> that is low memory.</p> <p>This function works like C-standard <code>printf</code> except that it only accepts <code>d</code>, <code>x</code>, <code>s</code>, <code>u</code> and <code>c</code> format specifiers with no conversions.</p> <p>The <code>p</code> format specifier is treated the same as a <code>x</code>.</p> <p>The capital version of each format specifier performs the same as the lower case equivalent.</p> <p>Any alignment or padding characters are simply ignored.</p> <p>The function uses the functions from <code>print.h</code> to do the underlying printing.</p> <p>Unlike <code>printf</code> this function has no return value.</p> <p>Whether the function does any output can be controlled via defines such as <code>DEBUG_PRINT_ENABLE</code> or <code>DEBUG_PRINT_ENABLE_[debug unit name]</code> in the application's <code>debug_conf.h</code></p>
Type	<code>void debug_printf(char fmt[], ...)</code>

### 1.2 Debug Units

A source file can be added to a debug unit by defining the `DEBUG_UNIT` macro before inclusion of `debug_print.h`. For example:

```
#define DEBUG_UNIT ETHERNET_MODULE
#include "debug_print.h"
```

To include all source files in a module in a particular debug unit, it is convenient to do it in the `module_build_info` file of the module e.g.:

```
MODULE_XCC_FLAGS = $(XCC_FLAGS) -DDEBUG_UNIT=ETHERNET_MODULE
```

If no `DEBUG_UNIT` is defined then the default debug unit is `APPLICATION`.

### 1.3 Enabling Printing

By default, debug printing is turned *off*. To enable printing you need to pass the correct command line option to compilation. The following defines can be set by using the `-D` option to the compiler. For example, the following in your application `Makefile` will enable debug printing:

```
XCC_FLAGS = ... -DDEBUG_PRINT_ENABLE=1
```

The following defines can be set:

**DEBUG\_PRINT\_ENABLE** Setting this define to 1 or 0 will control whether debug prints are output.

**DEBUG\_PRINT\_ENABLE\_[debug unit]** Enabling this define will cause printing to be enabled for a specific debug unit. If set to 1, this will override the default set by `DEBUG_PRINT_ENABLE`.

**DEBUG\_PRINT\_DISABLE\_[debug unit]** Enabling this define will cause printing to be disabled for a specific debug unit. If set to 1, this will override the default set by `DEBUG_PRINT_ENABLE`.

## APPENDIX A - Known Issues

There are no known issues with this library.

## APPENDIX B - Logging change log

### B.1 2.1.0

- ADDED: Now supports the %p format specifier
- CHANGE: Ignore the case of the format specifiers
- CHANGE: Ignore padding and alignment characters

### B.2 2.0.1

- CHANGE: Update to source code license and copyright

### B.3 2.0.0

- CHANGE: Restructured library