

How to use GCCE and Open C

There can be a problem when compiling an Open C application containing the main() entry point and using the GCCE target compiler. A workaround solution is described here:

Patch the file `\epoc32\tools\compilation_config\gcce.mk` file with the following:

```
AR=arm-none-symbianelf-ar
ARCHIVER=$(AR)
export ARCHIVER
```

Include a header called `staticlibinit_gcce.h` once within the application source code. Modify the MMP file with the following statements: `STATICLIBRARY libcrt0.lib`

Build the application using the GCCE target like **abld build gcce urel**.

The simple helloworld code which does this may look like the following:

```
#include <stdio.h>

#ifdef __GCCE__
// The following line is necessary in one file
#include <staticlibinit_gcce.h>
#endif

int main(void)
{
    printf("Hello World\n");
    return 0;
}
```

Compilation with GCCE

- Compilation with [GCCE](#) : warning: command line option "-Wno-ctor-dtor-privacy" is valid for C++/ObjC++ but not for C
- Since, [Open C](#) headers are located in `\epoc32\include\stdapis`,

"`SYSTEMINCLUDE \epoc32\include\stdapis`" must be included in the MMP file.

But wait! The above include doesn't seem to help. Instead, as in the first section on this page, patch the file `\epoc32\tools\compilation_config\gcce.mk` file, this time with the following:

```
# Moved "-Wno-ctor-dtor-privacy" to CPP_LANG_OPTION below
CC_WARNINGS_CONTROL_OPTION=-Wall -Wno-unknown-pragmas
```

and further down the file:

```
CPP_LANG_OPTION=-x c++ -Wno-ctor-dtor-privacy
```

This silences the warning message when compiling ANSI C code.

Internal Links

[Compiling C-code in UREL mode](#)