



[GNU Make] Makefile 教學：如何建立多設定檔



守不

開源教學 (<https://opensourcedoc.com/>)
/ [Makefile \(https://opensourcedoc.com/gnu-make/\)](https://opensourcedoc.com/gnu-make/)
/ 如何建立多設定檔專案



018

度也會逐漸拉長，若再加上跨平台的需求，設定檔，`make` 命令稿不僅限於單一檔案，我們可以將 `make` 命令稿拆成多個檔案，每一份 `make` 命令稿看起來就不會那麼冗長。一個常見的策略是將共通的部分保留在主要的 **Makefile** 中，再針對不同平台各自撰寫 `make` 設定檔。

多設定檔專案的前提在於我們可以從 **Makefile** 中呼叫其他的 **Makefile**。在終端機呼叫特定 `make` 命令稿的指令如下：

```
$ make -C /path/to/config -f config_file task
```

贊助商連結

贊助商連結

• • •

5.3折

熱銷破萬組明星經典款 DN...

5.9折 4.5折

DNA晶鑽...

5.3折

特潤活膚...

5.2折

頂級松露...

Makefile 中可參考以下方式：

someTask:
\$(MAKE) -C \$(TARGET_DIR) -f \$(CONFIG_FILE) task

在此處，我們用 `$(MAKE)` 變數而非實際的 `make` 指令，因為有時候 **Make** 的執行檔名稱是其他的名字，像 Windows 上的 **GNU Make** 移植品可能是 `mingw32-make`，而非 `make`。

5.9折

5.3折

5.3折

5.6折

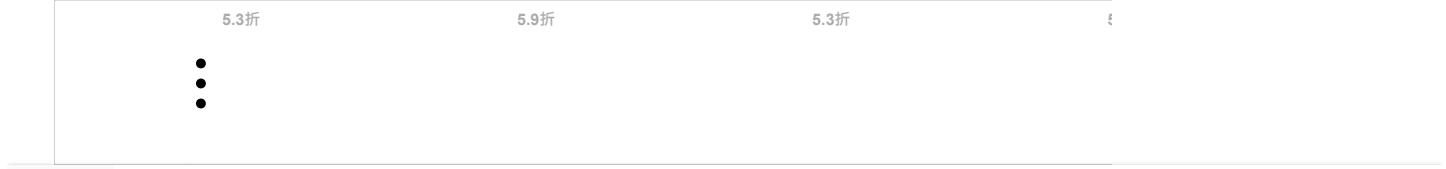
Unity Pro

Made with Unity built succ

Learn n

Hardspace: Shi by Blackbird In

unity



本專案的架構如下：

\$ tree

Makefile

src

(省略一些訊息)

project/main-content

Makefile

Makefile.win

test

Makefile

略去和 Make 設定檔無關的部分。由此輸出可看
定檔。

- make 或 make dynamic：編譯動態函式庫
- make static：編譯靜態函式庫
- make test：編譯並執行測試程式
- make memo：編譯測試程式後，檢查其記憶體使用情形
- make trim：去除檔案尾端的空白
- make clean：去除由編譯器所產生的檔案

贊助商連結

...

5.3折

熱銷破萬組明星經典款
DNA...

5.9折 4.5折

DNA晶鑽...

5.3折

特潤活膚...

5.2折

頂級松露...

本微專案在 Windows 和 GNU/Linux 上測試過，可使用 Visual C++、GCC、clang 等 C 編譯器來編譯此專案的程式碼。

在專案根目錄的 *Makeiflle* 是主要的 Make 設定檔，執行 Make 時會先取讀此設定檔。

一開始是偵測專案所在平台的程式碼：

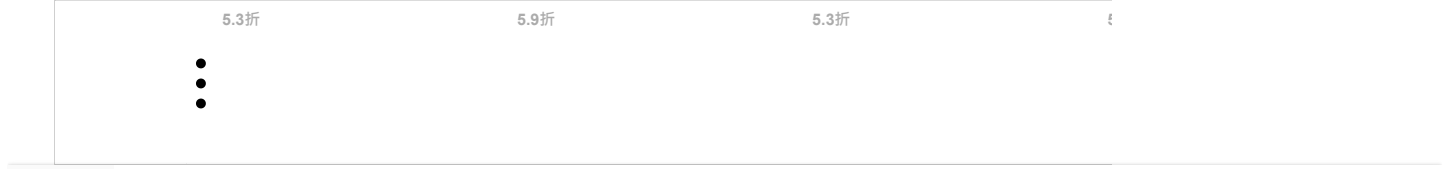
自己的婚戒自己做
WEDDING CODE 金工坊有多種款式，可製作體驗及精美飾品。
Wedding Code

```
ifeq ($(OS),Windows_NT)
    detected_OS := Windows
else
    detected_OS := $(shell sh -c 'uname -s 2>/dev/null || echo not')
endif


export detected_OS
```

贊助商連結

Unity Pro
Made with
Unity
built
succ
Learn n
Hardspace: Shi
by Blackbird In
unity



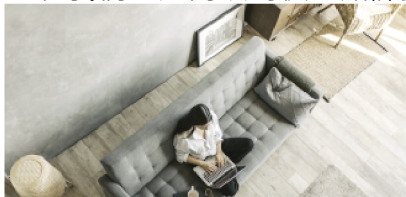
我們動態地設定 **Make** 設定檔名稱：



```
ifeq ($(detected_OS),Windows)
    CONFIG=Makefile.win
else
    CONFIG=Makefile
endif
```

回頁頂 (/gnu-make/multi-file-project/#main-content)
之後就不需要把不同系統所用的設定檔名稱寫死在設定檔中。在此專案中，類 **Unix** 系統共用一個設定檔，**Windows** 系統使用一個設定檔。

在我們上述的改寫後，編譯動態函式庫的指令如下：



```
R) -f $(CONFIG) dynamic
```

乍看是同一條指令，在不同系統上實際對應的指令會動態改變。我們就是透過這條指令呼叫相對應的 **Make** 設定檔並執行相關的任務。

本專案執行測試的指令如下：

```
test: trim
    $(MAKE) -C $(SOURCE_DIR) -f $(CONFIG) compile_debug
    $(MAKE) -C $(TEST_DIR) -f $(CONFIG) test

trim:
    $(MAKE) -C $(SOURCE_DIR) -f $(CONFIG) trim
    $(MAKE) -C $(TEST_DIR) -f $(CONFIG) trim
```

為什麼要這樣寫呢？我們想在執行 **test** 任務前，先將程式碼尾端的空白去掉，所以我們在 **test** 任務前會呼叫 **trim** 任務。

在這個專案中，程式碼分散在兩處，所以我們要先在 `$(SOURCE_DIR)` 中編譯目標檔 (objects)，再到 `$(TEST_DIR)` 編譯並執行測試測試程式。

從這些指令可看出，我們的指令基本上和平台脫勾了，實際的指令會根據不同的 **Make** 設定檔而有所不同。接著，我們以 `src/Makefile.win` 為例，來看實際的指令如何撰寫。

我們動態地決定 **CFLAGS**：

贊助商連結



5.3折

...

熱銷破萬組明星經典款

DNA...

5.9折

4.5折

DNA晶鑽...

5.3折

特潤活膚...

5.2折

頂級松露...

贊助商連結



Unity Pro

Made with

Unity built succ

Learn n

Hardspace: Sh by Blackbird In

unity

```
ifeq ($(TARGET),Debug)
    CFLAGS=$(CFLAGS_DEBUG)
else
```

```

return 0;
}
#endif
#endif
project/#main-content)

```

[回首頁 \(1\)](#)

我們根據 C 編譯器的不同，決定目的檔的名稱：

```
r.obj bstnode.obj
```

```
o bstnode.o
```

的不同，動態地設定函式庫檔名：

```

ifeq ($(CC),cl)
    DYNAMIC_LIB=algobstreei.dll
else
    DYNAMIC_LIB=libalgobstreei.dll
endif

```

```

ifeq ($(CC),cl)
    STATIC_LIB=algbostreei.lib
else
    STATIC_LIB=libalgbostreei.a
endif

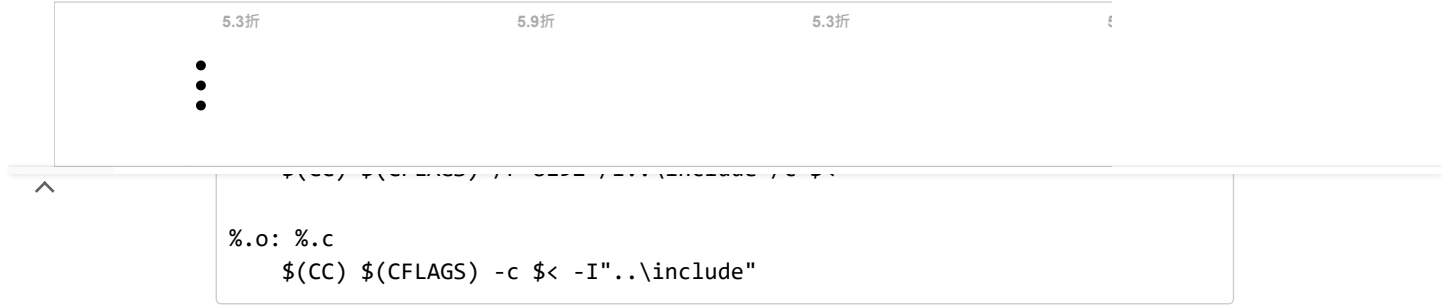
```


我們根據 C 編譯器的不同，決定編譯動態函式庫的指令：

```
dynamic:
ifeq ($(CC),cl)
    for %x in (*.c) do $(CC) $(CFLAGS) /F 8192 /I..\include /c %x
    link /DLL /out:$(DYNAMIC_LIB) *.obj
else
    for %x in (*.c) do $(CC) $(CFLAGS) -fPIC -c %x -I"..\\include"
    $(CC) $(CFLAGS) -shared -o $(DYNAMIC_LIB) *.o -I"..\\include"
endif
```



在此段程式碼中，我們假定使用者使用 Visual C++ 或 MinGW，考量目前 C 編譯器的市佔率，這樣的寫法應可滿足大部分的使用者。

在編譯給測試程式的目的檔時，我們刻意指定適合除錯的編譯器參數：





 將 Make 命令稿分離，我們可以減少單一 Make 命令稿的長度，並以更有邏輯的方式組織 Make 命令稿。

分享本文 /gnu-make/multi-file-

 Facebook (<https://www.facebook.com/sharer.php?u=https%3a%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f>)  Twitter (<https://twitter.com/intent/tweet?url=https%3a%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f>)



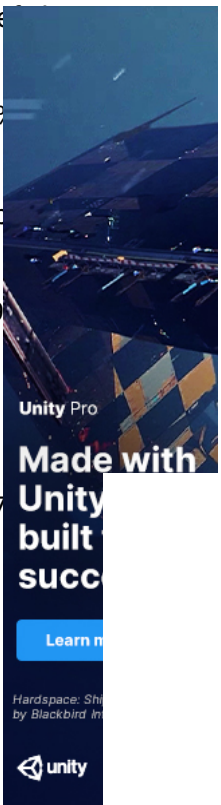
 Skype (<https://web.skype.com/share?url=https%3a%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f&text=%5bGNU%20Make%5d%20Makefile%20e6%95%99e5%ad%b8%ef%bc%9a%e5%a1>)  EverNote (<http://www.evernote.com/clip.action?url=https%3a%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f&text=%5bGNU%20Make%5d%20Makefile%20e6%95%99e5%ad%b8%ef%bc%9a%e5%a1>)

贊助商連結



 Email (<https://mail.google.com/mail/?ui=cm&to=&su=%5bGNU%20Make%5d%20Makefile%20e6%95%99e5%ad%b8%ef%bc%9a%e5%a1%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f>)  Yahoo (<http://compose.mail.yahoo.com/?subject=%5bGNU%20Make%5d%20Makefile%20e6%95%99e5%ad%b8%ef%bc%9a%e5%a1%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f>)  Facebook (<https://www.facebook.com/%E9%96%8B%E6%94%BE%E5%8E%9F%E5%A7%8B%E7%9A%A7%2f%2fopendsourcedoc.com%2fgnu-make%2fmulti-file-project%2f>)  Twitter (<https://twitter.com/openSourceDok>)

贊助商連結





0則留言

開源技術教學網

Disqus 隱私政策

Favorite

Tweet

分享



DISQUS 帳號

通過以下方式登錄

或注冊一個 DISQUS 帳號 ?

回頁頂 (/gnu-make/multi-file-

project/#main-content)

姓名

回首頁 (/)

Copyright (c) 2014-2022 Michelle Chen. All Rights Reserved.



贊助商連結

5.3折

熱銷破萬組明星經典款
DN...

5.9折 4.5折

DNA晶鑽...

5.3折

特潤活膚...

5.2折

頂級松露...

贊助商連結

Unity Pro

Made with
Unity
built
succ

Learn n

Hardspace: Shi
by Blackbird In

unity