

附錄 A

Java 2 SDK 版原始碼概觀

■ 簡介

據說資訊社會的演進有個 lifecycle，從早期每個人的辦公桌上只有一部終端機的時代慢慢進步到 PC 時代，然後我們發現逐漸地潮流又走向 NC(Network Computer)，廠商鼓吹 thin client 的時代。對於原始碼的看法亦是如此，從早期廠商在出售的機器內附上程式原始碼供顧客自行修改，然後漸漸地程式碼變成廠商獨有而不願公開的資產，最後又演變成大家鼓吹 Open Source、Free Software 的時代。姑且不論普羅大眾是否真的有興趣去 trace 附在您電腦之中的原始碼，但是我想這就是開放原始碼最可貴的地方 – 大家都有機會一窺系統的內部行為，自己電腦上資料的安全以及程式的運作方式再也不會掌控在特定廠商的手上，有了原始碼，我們在也不用懷疑是否某個廠商的作業系統或是應用程式會偷偷地在背後將您的資料回傳到他們公司的伺服器中。

隨著這股開放原始碼的潮流，Java2 SDK 的原始碼其實已經 release 出來很長一段時間了，但是採用的 License 為 Sun Community Source Code License(SCSL)，與一般我們討論到 Open Source 時所會提到的 General Public License(GPL)有所不同。相信最近大家會發現，在相關文章中也開始提及將 Java SDK 改用 General Public License 釋出的議題。當 2000 年 10 月 Sun 將 Star Office 的原始碼以 General Public License 釋出之後，接下來大家所期待的就是 Sun 將 Java SDK 的原始碼以 General Public License 釋出，如果此事成真，那麼 Java 今後的發展將不是我們目前所能預測的，如果成功，那麼 Java 會像 Linux 一樣展翅高飛，如果沒有成功，那麼將可能像 Mozilla 計劃一般雷聲大雨點小。

先姑且不論哪個 License 比較 open，哪個比較 free，筆者撰寫此系列文章的目的只有一個，就是挖掘出存在於 Java2 SDK 內部的運作機制。如果您希望了解一些我們常用的執行檔(exe，例如 javac.exe、java.exe 等)、幾個重要的動態連結函式庫(dll，例如 jvm.dll、java.dll 等)，以及其他使用 Java 程式語言所撰寫的套件(package，例如 java.io、java.util 等)，最終都需要從原始碼之中一探究竟。

當然，trace 別人的程式碼這種事情乍看之下有點像是程式黑手在做的事情，但是筆者深深覺得，我們的軟體工業要站上國際舞台仍然需要很長一段時間。從大觀點來看，系統分析、系統設計、系統測試、如何妥善地分配有限人力這些議題上，至少我們還沒有辦法舉出哪些國內公司將這些工作做的十分出色，從 Java 2 SDK 的原始碼結構中，我們可以看到開發一套 SDK(Software

Development Kit)時採用的團隊組織架構之縮影，原始碼的位置怎麼放？那麼多程式設計師寫出來的程式如何整合在一起發揮縱效？如何讓文件的產生自動化？我想我們一定可以從 Java 2 SDK 的原始碼得到一些答案。從小觀點來看，如何善用 Design Pattern？如何寫出有效率的程式碼？如何讓程式設計師寫出來的程式碼具有一致性？我們都知道 Java 內部的套件大量地使用了 Design Pattern，我們也知道 Java 因為虛擬機器的關係，所以程式碼必須非常地有效率，相信這些問題也能夠從 Java 2 SDK 的原始碼獲得答案。

大力推展 Open Source 的開發中國家也非常清楚，開放原始碼將是開發中國家趕上已開發國家軟體工業的終南捷徑。期待個人的努力可以成為這個大革命中一顆不可或缺小螺絲釘。

在本附錄中，筆者將為大家介紹 Java 2 SDK 原始碼的輪廓，讓每位讀者都具有取得 Java 2 SDK 原始碼並能夠成功編譯這些原始碼的基本能力。廢話不多說，我們就開始囉。。

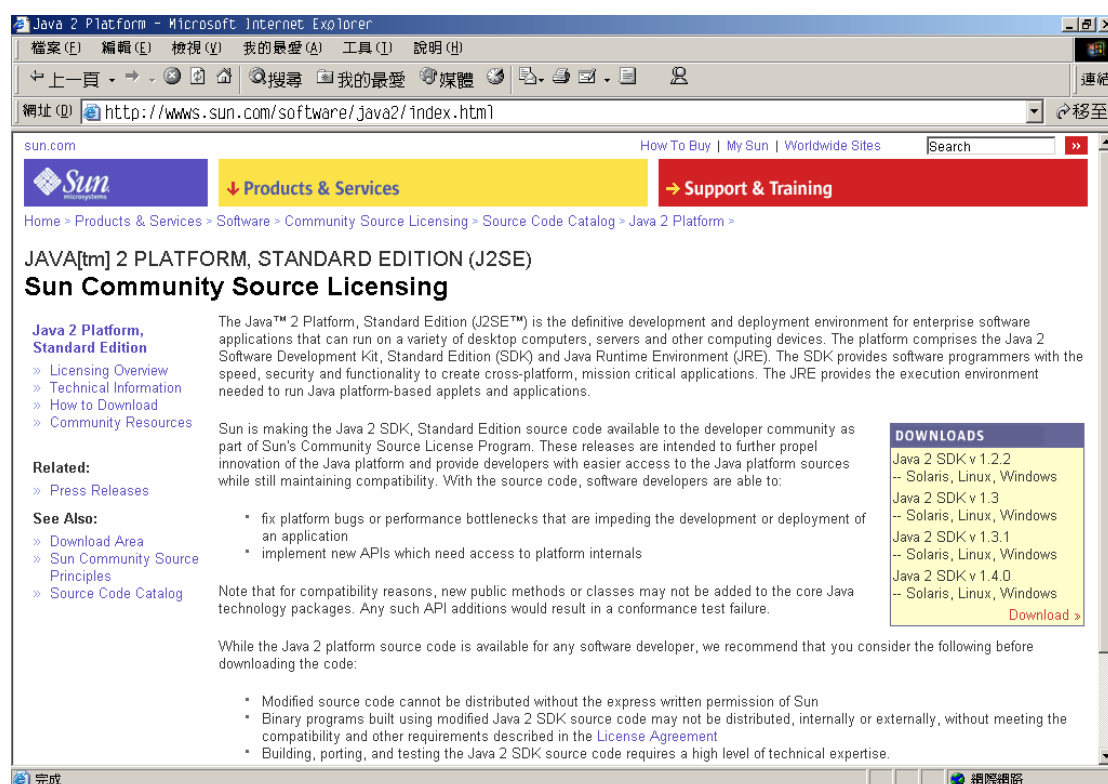
如何取得 Java2 SDK 原始碼

要取得 Java2 SDK 的原始碼，首先您必須要是 Java Developer Connection(JDC) 的一員，所以請您先到 <http://developer.java.sun.com/developer/> 網站上註冊，取得使用者 ID 和密碼。如下圖所示：

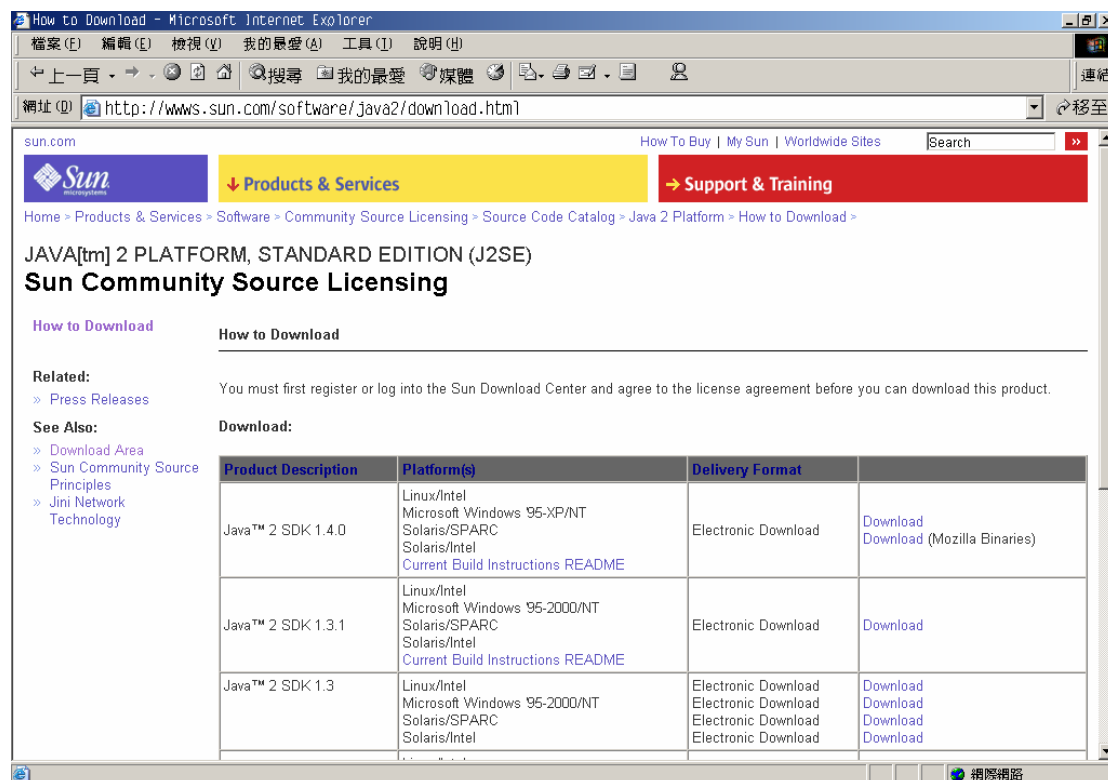


當您成功成為 Java Developer Connection(JDC)的一員之後，請您到 <http://www.sun.com/software/communitysource/java2/>，您將會在右手

邊看到下載 Java 2 SDK 原始碼的選項:



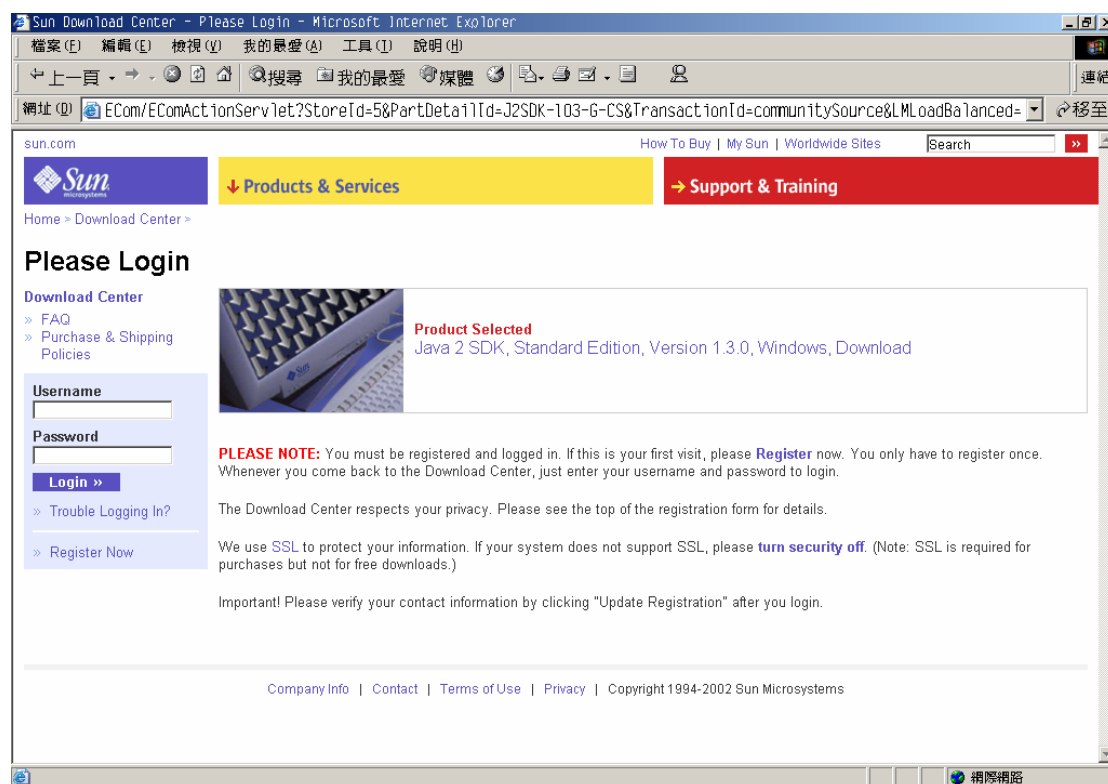
現在提供 Java 2 SDK 1.2.2、1.3、1.3.1、以及 1.4.0 版本原始碼的下載。請點選右邊紅色的“Download”。



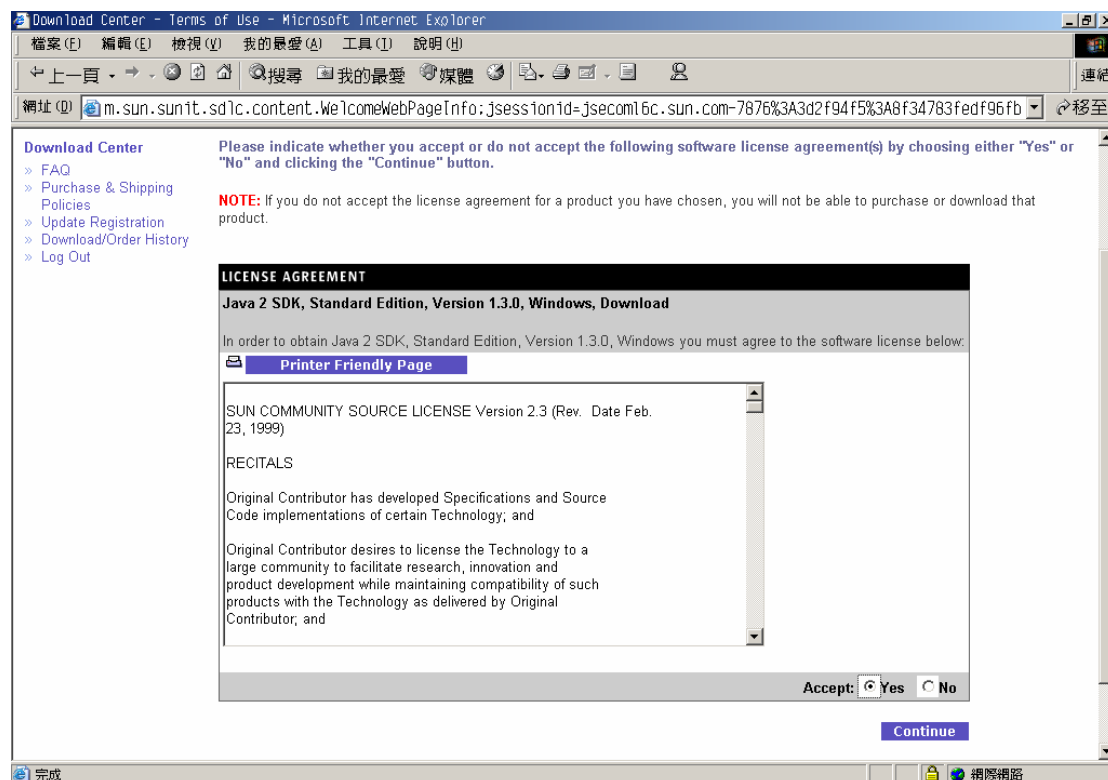
請點選您希望下載的版本，並點選“Download”。您可以發現，在 Java 2

SDK 1.3.1 以上的版本，都已經可以直接從網路上觀看原始碼的編譯方式，而之前的版本則必須等到下載之後，才能找到如何編譯原始碼。

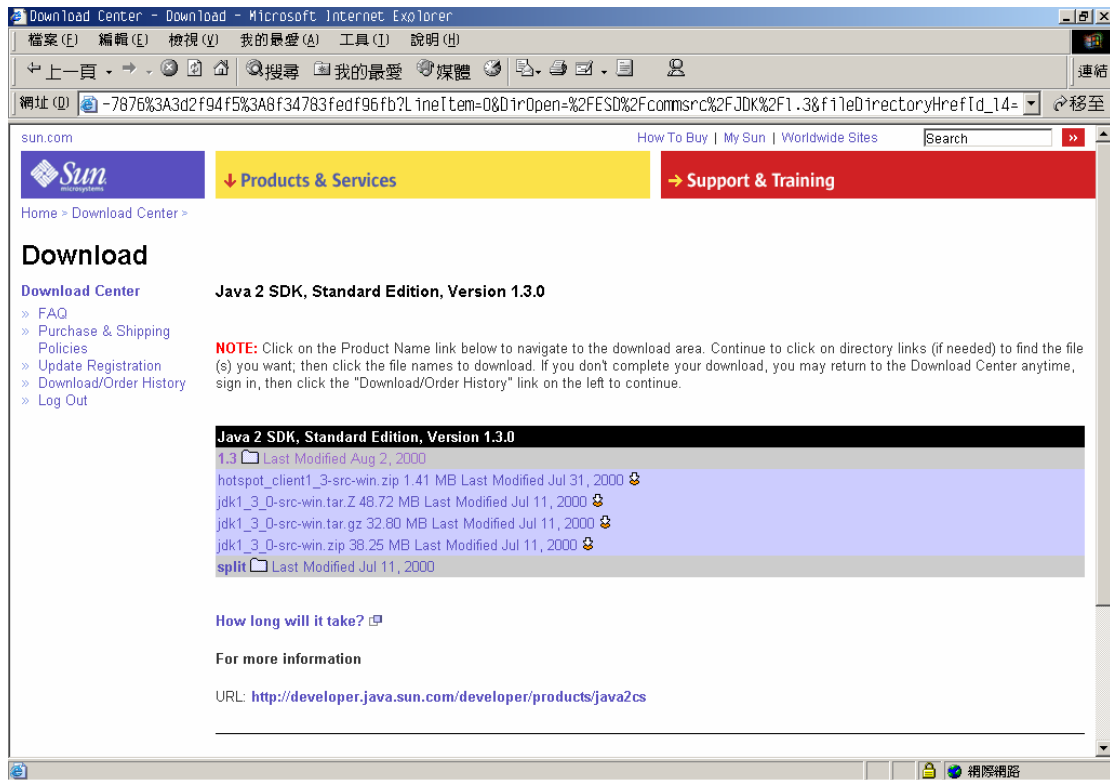
假設您選擇下載 1.3.0，則接下來的畫面如下：



請在左方輸入您在 Java Developer Connection(JDC)的使用者 ID 和密碼，接下來則出現版權宣告畫面：



如果您接受版權宣告的內容，請在 Accept 處選擇 Yes，然後按下 Continue，就開始選擇 Java 2 SDK 原始碼的動作。如下圖所示：

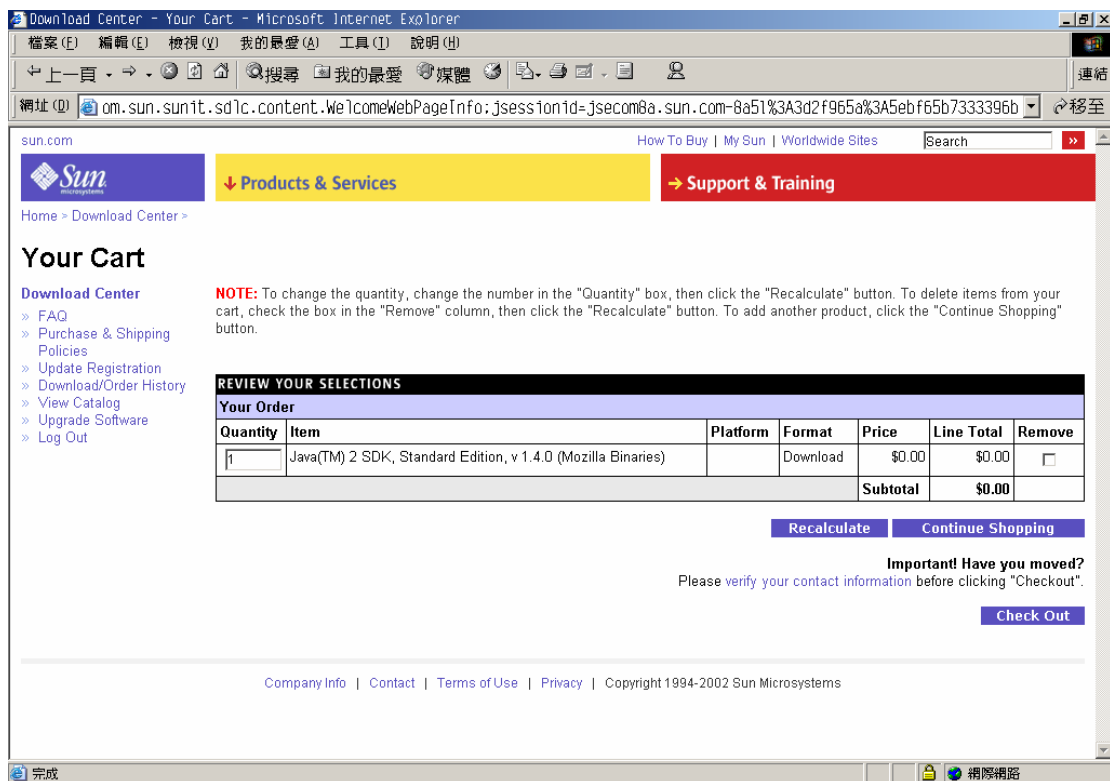
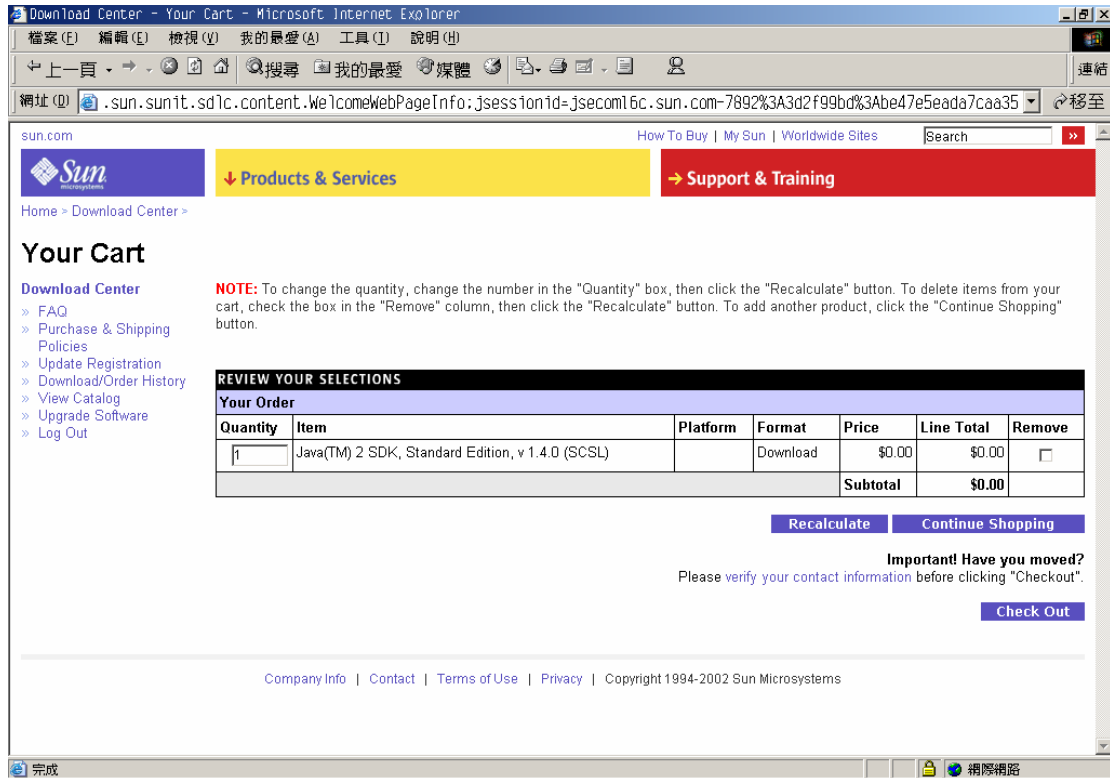


請選擇您希望下載的版本，就開始下載程序。

如果您選擇下載 Java 2 SDK 1.4.0 的原始碼，則程序上稍有不同。由 Java 2 SDK 1.4.0 內附可以支援 Netscape 或 Mozilla 瀏覽器的 Plug-in，所以編譯時必須有 Mozilla 的相關檔案才能完整編譯。因此，請您除了下載 Java 2 SDK 1.4.0 的原始碼之外，還必須額外下載 Mozilla Binaries。

Product Description	Platform(s)	Delivery Format	
Java™ 2 SDK 1.4.0	Linux/Intel Microsoft Windows 95-XP/NT Solaris/SPARC Solaris/Intel Current Build Instructions README	Electronic Download	Download Download (Mozilla Binaries)

底下我們同時列出兩個下載的畫面：



如上圖所示，螢幕上會出現類似購物車的畫面，不過請放心，不需要付錢 ☺。請點選右下角的“Check Out”即可。接下來一樣會出現版權宣告畫面，請一樣在 Accept 處選擇 Yes，然後按下“Continue”即可。

最後，會出現類似網路購物結帳的畫面，按下右下角的“Place Order”就可

以繼續下載動作。

Download Center - Checkout - Verify Order - Microsoft Internet Explorer

Home > Download Center >

Verify Order

Download Center

- » FAQ
- » Purchase & Shipping Policies
- » Log Out

Billing Address		Payment Method			
Mr. Wang Sen 3F, No 223, Gong Yuan RD Hsin Chu, Taiwan, Taiwan		No Payment Required			
Edit					
Quantity	Item	Platform	Format	Unit Price (USD)	Line Total (USD)
1	Java(TM) 2 SDK, Standard Edition, v 1.4.0 (SCSL)		Download	\$0.00	\$0.00
Change Order				Subtotal	\$0.00
				Sales Tax	\$0.00
				Total (USD)	\$0.00

IMPORTANT: Please confirm all the information above is correct before proceeding. A copy of the receipt will be e-mailed to you.
If you would like to send a copy of the receipt to a different e-mail address, please enter it here: (no group aliases please)

[Place Order](#)

Company Info | Contact | Terms of Use | Privacy | Copyright 1994-2002 Sun Microsystems

Download Center - Checkout - Verify Order - Microsoft Internet Explorer

Home > Download Center >

Verify Order

Download Center

- » FAQ
- » Purchase & Shipping Policies
- » Log Out

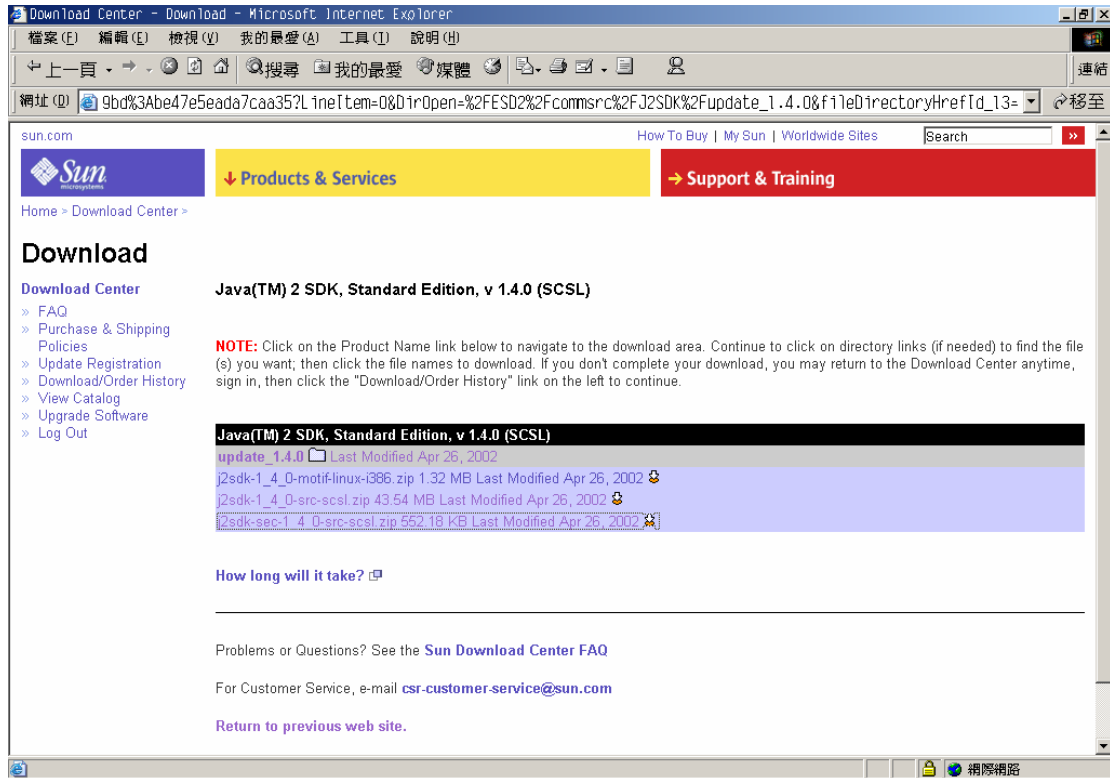
Billing Address		Payment Method			
Mr. Wang Sen 3F, No 223, Gong Yuan RD Hsin Chu, Taiwan, Taiwan		No Payment Required			
Edit					
Quantity	Item	Platform	Format	Unit Price (USD)	Line Total (USD)
1	Java(TM) 2 SDK, Standard Edition, v 1.4.0 (Mozilla Binaries)		Download	\$0.00	\$0.00
Change Order				Subtotal	\$0.00
				Sales Tax	\$0.00
				Total (USD)	\$0.00

IMPORTANT: Please confirm all the information above is correct before proceeding. A copy of the receipt will be e-mailed to you.
If you would like to send a copy of the receipt to a different e-mail address, please enter it here: (no group aliases please)

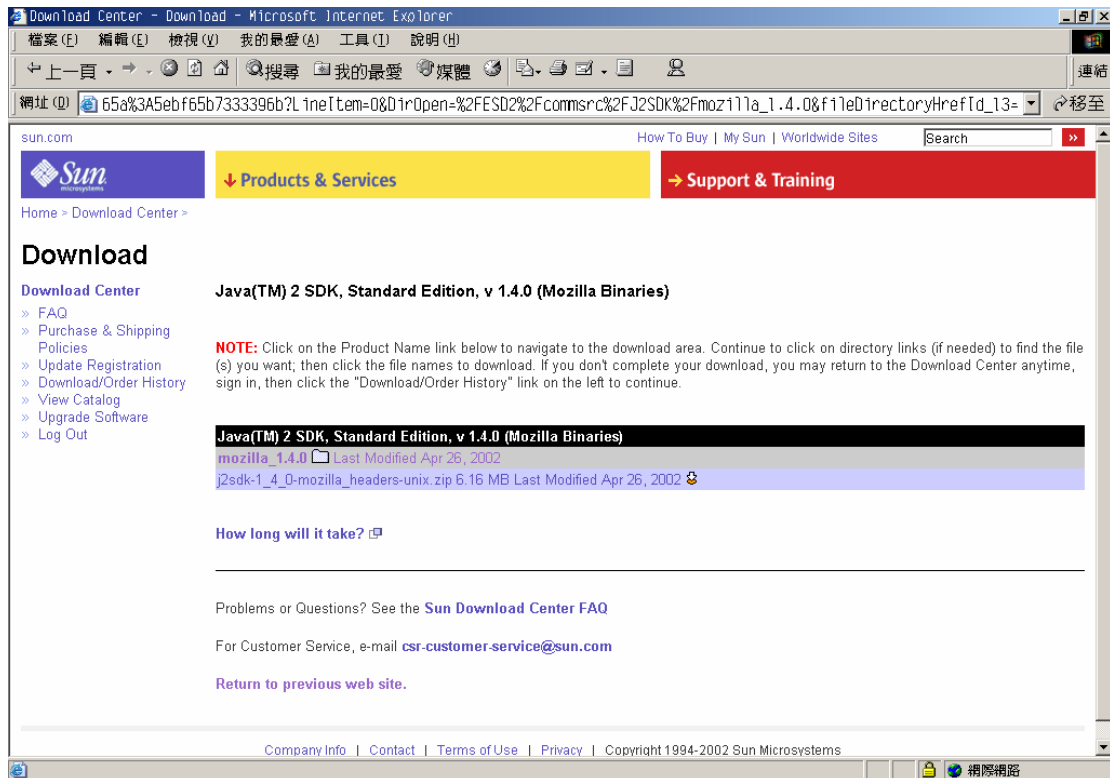
[Place Order](#)

Company Info | Contact | Terms of Use | Privacy | Copyright 1994-2002 Sun Microsystems

最後請選擇您欲下載的檔案:

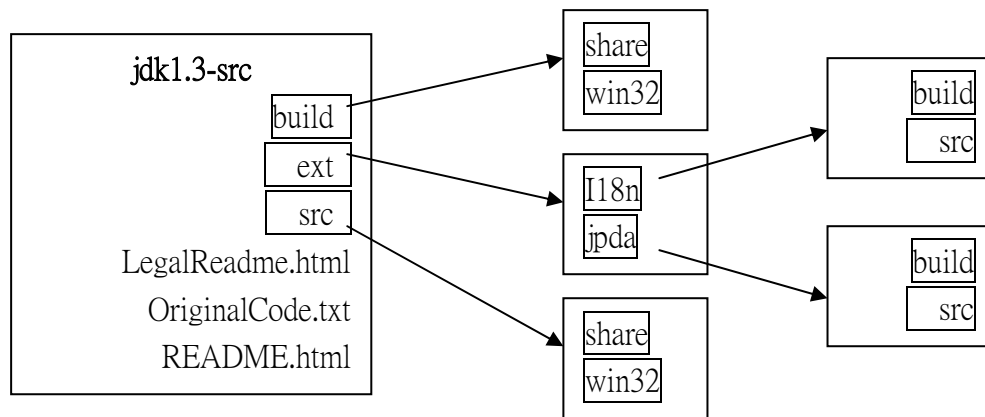


如果您只需要 Java 2 SDK 1.4.0 的原始碼，那麼只要下載 j2sdk-1_4_0-src-scs1.zip 即可，而 j2sdk-sec-1_4_0-src-scs1.zip 是 Java Cryptography Extension 的原始碼，不下載也不會影響編譯程序。



Java 2 SDK 1.3.0 原始碼的架構

當您解開 Java 2 SDK 1.3.0 版原始碼的壓縮檔之後，您會看到如下圖所示的目錄結構：



◇ build 目錄

放置了所有的 makefile 檔，這些 makefile 為 Visual Studio 內附的 nmake.exe 所能解讀的格式。通常 makefile 的檔名有五種，分別是 Makefile、*.nmk、*.jmk(指定要編譯的 .java 檔)、*.cmk(指定要編譯的 .c 檔)、*.mk。在 build\share 底下您還可以找到一些 .java 檔。

由於我們要編譯的是 Windows 作業系統下的 Java SDK，所以 Makefile 的主檔為 build\win32\Makefile，整個 Java 2 SDK 的編譯流程都是根源於這個檔案，在 build\win32\makefiles 目錄之下您可以找到共用的 Makefile，通常附檔名為 .nmk。

build\win32 目錄下的 makefile 都是用來建造和 Windows 作業系統相關的程式。而在 build\share 目錄底下就多為 .mk、.jmk、.cmk 檔，這些檔案通常用來建造和 Windows 作業系統無關的程式。

◇ ext 目錄

這個目錄底下放置了用來支援 Java 2 SDK 函式庫的 makefile 與原始碼，這些原始碼皆使用 Java 程式語言所撰寫。

ext\i18n 目錄下放置了與國際化(internationalization)相關的函式庫，這些編譯過的檔案最後都被放到 i18n.jar 裡頭。

ext\jpda 目錄下放置了與 Java 平台除錯器架構(Java Platform Debugger Architecture)相關的檔案。

◇ src 目錄

這個目錄放置了所有建構 Java SDK 所需要用到的 .c 檔(以 C 程式語言撰寫)以及 .java 檔(以 Java 程式語言撰寫)。

src\win32 目錄下都是用來建造和 Windows 作業系統相關的原始碼。而在 src\share 目錄底下就多為與 Windows 作業系統無關的原始碼。因此聰明的讀者們應該可以猜到大部分 Java 標準套件的原始碼大多放在 src\share 目錄之下。這些標準套件的原始碼在最後都被加入到 src.jar 之中，而編譯過的 .class 檔則被打散放入許多不同的 jar 檔內。

◇ LegalReadme.html

Sun Community Source Code License 版權宣告

◇ OriginalCode.txt

Java2 SDK 原始碼中所有檔案的列表。

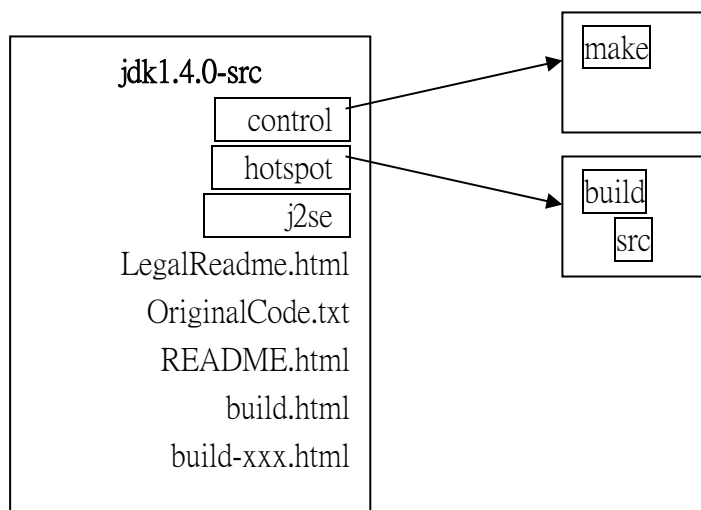
◇ README.html

關於 Java 2 SDK 原始碼的說明，該內容大部分都已經被筆者吸收而整合到這篇附錄之中。

從 Java 2 SDK 原始碼的架構中，我們可以發現其檔案的放置方法很獨特，是採用 makefile 與原始碼分開放置的做法，不同於一般我們所見(例如 Linux kernel 原始碼)採 makefile 和原始碼放在一起的做法。這兩種做法各有利弊。以 Linux Kernel 原始碼的做法，每次編譯完成之後，您將看到原始碼檔案和編譯過程中產生的中間檔混雜在一起，感覺有點混亂。如果我們採用 Java 2 SDK 原始碼的目錄結構，每次編譯出來的中間檔和原始碼就會分隔在不同的目錄下，看起就非常賞心悅目，但是，這樣放置的代價就是 makefile 裡頭對於路徑的參照變得十分複雜，您將會在 Java 2 SDK 原始碼的 makefile 中發現很多 ..\..\..\src 的敘述，看起來真是不舒服，如果我們要了解該執行檔是由哪些檔案構成的時候就很辛苦了。

■ Java 2 SDK 1.4.0 原始碼的架構

當您解開 Java 2 SDK 1.4.0 原始碼的壓縮檔之後，您會看到如下圖所示的目錄結構，與 1.3.0 版其實有蠻大的不同：



◇ control 目錄

放置了建夠整個 Java 2 SDK 裡面每一個組成成分所共享的 makefile 檔，而且附檔名都是.gmk，所以這些共享的 makefile 與特定平台無關。而其他的 makefile 則藏身在 hotspot\build 目錄底下，這個目錄底下分別有編譯 linux、solaris 以及 Windows 平台上之 Java 虛擬機器所需的 makefile。在 j2se\make 目錄底下也放置著許多 makefile，這些 makefile 都是用來建夠 Java 2 SDK 內的每一個組成分子，比方說 Java 類別函式庫、Java 2 SDK 內附的開發工具等等。

◇ hotspot 目錄

內含 Hotspot Client VM 與 Hotspot Server VM 的原始碼以及 makefile。

◇ j2se 目錄

內含 Java 2 SDK 內的每一個組成分子的原始碼與 makefile。比方說 Java 類別函式庫、Java 2 SDK 內附的開發工具等等。

◇ LegalReadme.html

Sun Community Source Code License 版權宣告

◇ OriginalCode.txt

Java2 SDK 原始碼中所有檔案的列表。

◇ README.html

關於 Java 2 SDK 原始碼的說明，該內容大部分都已經被筆者吸收而整合到這篇附錄之中。

◇ build.html 、 build-linux.html 、 build-solaris.html 、 以及 build-win32.html

說明如何編譯各個平台上 Java 2 SDK 的文件。

從上述說明我們可以看出，Java 2 SDK 1.4.0 的原始碼結構比起 Java 2 SDK 1.3.0 的原始碼結構要更清楚，要深入每一個原始碼也因此更方便。

底下我們以 Java 2 SDK 1.3.0 的原始碼為例，說明如何完成其編譯程序。如果您想編譯 Java 2 SDK 1.4.0 版，請多參考 build-win32.html，裡面對於編譯程序講的十分清楚。

■如何編譯 Java 2 SDK 1.3.0 原始碼

當我們深入了解 Java 2 SDK 原始碼之後，很多工程師必定想自己嘗試修改原始碼，然後進行編譯，我想這是身為一個工程師最大的樂趣。當然，在我們還沒有修改之前，我們也必須試著編譯整個原始碼，順便測試一下原始碼是否完整，因此接下來要介紹編譯 Java 2 原始碼的準備步驟和方法。

要順利編譯 Java2 SDK 原始碼，您必須符合底下幾種條件：

(1) 作業系統

您必須在 Windows NT 4.0 或是 Windows 2000 上才能順利編譯 Java 2 原始碼。一旦編譯完成之後，所產生的執行檔或是動態連結和式庫將能夠在所有的 Win32 作業平台上執行。

(2) 硬體設備

在 Java2 SDK 原始碼內附的文件之中，建議的硬體設備為 Pentium 等級的處理器，配上至少 128 MB 的記憶體。在筆者的電腦上，Pentium III 450 配上 PC 100 128 MB 的記憶體，總共花了將近 2.5 個小時完成所有編譯動作。如果算上編譯時期遇到的一些錯誤以及恢復錯誤的時間，大概花了 3 個小時。

(3) 開發工具

我們必須準備幾項工具，他們分別是：

- Microsoft Visual C++ version 6.0，並配上最新的 Service pack(或者 Visual Studio 97 配上 Service Pack 3)。

在 Java2 SDK 原始碼之中，存在有許多與 Windows 平台相關的程式碼，這些程式碼都以 C 語言撰寫而成，為了順利編譯這些原生程式碼，我們必須要借助 Visual C++ 編譯器以便產生能在 Windows 平台上執行的執行檔(.exe)以及動態連結和式庫(.dll)。

同時，Java2 SDK 原始碼之中有許多 Makefile，這些 Makefile 是用來控制所有程式的編譯動作。我們必須借助 Visual C++ 內附的 nmake.exe 這個工具以幫助我們解讀這些 Makefile 並順利編譯出整個 Java2 SDK。

- MKS Toolkit version 5.2 以上的版本。

MKS Toolkit 是一套能夠在 Windows 上模擬 UNIX 執行環境的工具，請到 <http://www.mks.com> 下載這套工具的 30 天試用版(正式版必須要付費)。或者如果您有訂閱 MSDN Profession 以上的版本，您可以在光碟中找到 Microsoft Windows Services for UNIX 2.0 Add-On Pack(在 UNIX20_INSERV\3RDPARTY\MKS 目錄底下，執行 x86 子目錄中的 setup.exe 即可安裝)。30 天試用版的功能騎士和付費版本的功能一樣，只不過當您在編譯 Java 2 SDK 原始碼的時候，只要用到了 MKS Toolkit 裡的工具，就會出現一個廣告畫面罷了，應該不會影響您的心情才是。

- Java 2 SDK 1.3 的可執行版本。

相信讀者們看到這個地方，一定會覺得很疑惑。奇怪，編譯 Java 2 SDK 的原始碼竟然也需要 Java 2 SDK 的協助？不禁讓人想起雞生蛋蛋生雞的問題，不是嗎？其實 Java 2 SDK 在整個編譯過程中，扮演的是 bootstrap compiler(靴帶式編譯器)的角色。之所以需要 Java 2 SDK 的原因是因為大家在編譯 Java 程式時所使用的 javac.exe 核心程式竟然也是用 Java 撰寫而成的，由於 Java

2 SDK 的原始碼有許多與平台無關、以 Java 程式語言所撰寫的程式，因此在編譯初期，我們必須借助過去舊版的 Java 2 SDK 或是 JDK 幫助我們產生新版的 Java 2 SDK，再利用新產生的新版 javac.exe 編譯 Java 2 SDK 原始碼內的其他 .java 檔。

當大家將所有必備的條件都準備妥當之後，接著我們要開始準備編譯前的相關工作。底下的內容將有下列假設：

工具	安裝路徑
Windows NT/2000 系統所在目錄	c:\winnt
Microsoft Visual C++ version 6.0	d:\Microsoft Visual Studio
Microsoft Visual Studio 97	d:\Program Files\DevStudio
MKS Toolkit	c:\mskdemo\mksnt
Java2 SDK 1.3 的可執行版本	d:\jdk1.3
Java2 SDK 原始碼	d:\jdk1.3-src

接下來我們要設定環境變數，為了方便起見，請大家新增一個叫做 env.bat 的批次檔，這樣以後要修改環境變數的時候就不必大費周章：

如果您使用的 C 編譯器為 Microsoft Visual C++ version 6.0，那麼 env.bat 的內容如下：

env.bat

```
set MSVCDir=d:\Microsoft Visual Studio\vc98
set MSDevDir=d:\Microsoft Visual Studio\Common\MSDev98
set WinDir=c:\winnt
set include=%MSVCDir%\include;%MSVCDir%\mfc\include;%MSVCDir%\atl\include
set lib=%MSVCDir%\lib;%MSVCDir%\mfc\lib
set
PATH=c:\mskdemo\mksnt;%MSVCDir%\bin;%MSDevDir%\bin;%WindowsDir%\system32;%PATH%
set ALT_BOOTDIR=d:\jdk1.3
set TMPDIR=d:\jdk1.3-src\tmp
```

如果您使用的 C 編譯器為 Microsoft Visual Studio 97，那麼 env.bat 的內容如下：

env.bat

```
set MSVCDir= d:\Program Files\DevStudio\VC
set MSDevDir= d:\Program Files\DevStudio\SharedIDE
set WinDir=c:\winnt
set include=%MSVCDir%\include;%MSVCDir%\mfc\include;%MSVCDir%\atl\include
```

```
set lib=%MSVCDir%\lib;%MSVCDir%\mfc\lib
set
PATH=c:\mskdemo\mksnt;%MSVCDir%\bin;%MSDevDir%\bin;%WindowsDir%\system3
2;%PATH%
set ALT_BOOTDIR=d:\jdk1.3
set TMPDIR=d:\jdk1.3-src\tmp
```

任何一個環境變數設定有誤，都會造成編譯錯誤，請讀者們務必小心。關於環境變數，底下有幾件事情請讀者一定要注意：

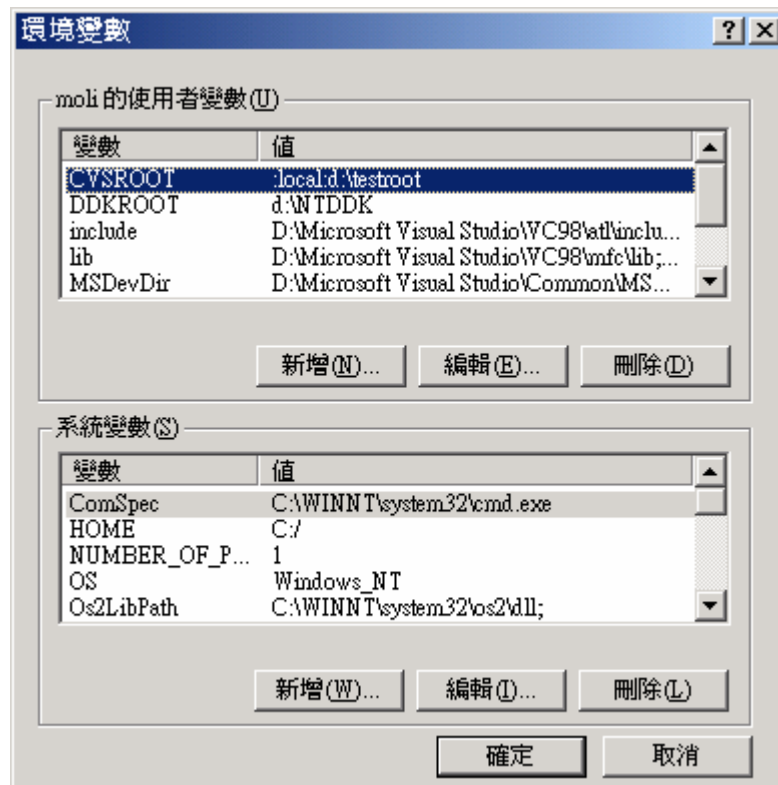
1. 對於 PATH 環境變數來說，MKS Toolkit 的路徑一定要出現在 Microsoft Visual C++ version 6.0(或是 Microsoft Visual Studio 97)的所在路徑之前，否則會造成編譯錯誤。所以請不要將 PATH 設定為：

```
set
```

```
PATH=%MSVCDir%\bin;%MSDevDir%\bin;c:\mskdemo\mksnt;%WindowsDir%\system3
2;%PATH%
```

(MKS Toolkit 出現在 Microsoft Visual C++ version 6.0(或是 Microsoft Visual Studio 97)的所在路徑之後)

2. 請關閉 CLASSPATH 環境變數。如果您是在 Windows 2000 底下，請在“我的電腦”圖示上按下滑鼠右鍵，選擇“內容”，點選“進階”次頁，再按下“環境變數”按鈕。將出現環境變數對話方塊，如下圖所示：



請尋找使用者變數中或是系統變數中任何名為 CLASSPATH 的環境變

數刪除，否則在編譯 Java 2 SDK 原始碼初期就會出現錯誤訊息。

3. 環境變數 ALT_BOOTDIR 是用來設定 bootstrap compiler(靴帶式編譯器)的所在位置，您只要設定為 Java SDK 的安裝位置就可以了。

當上述準備工作全部完成之後，請開啟“命令提示字元”視窗(就是 DOS 視窗啦!)，並將目錄切換到 d:\jdk1.3-src\build\win32 底下，在命列上輸入

nmake world

此時編譯 Java2 SDK 原始碼的過程正式展開，毫無疑問地，這將是一個漫長的過程。

在編譯的過程之中，在筆者的電腦上出現了一些編譯錯誤，在這裡順便向各位提及，順便說明筆者的解決辦法：

- 找不到 .properties 檔:
在編譯 Java2 SDK 原始碼的時候常常出現找不到 .properties 檔的錯誤訊息，如果發現編譯終止的原因是因為在個特定目錄底下找不到某個 .properties 檔，那麼請讀者自行搜尋 jdk1.3-src\src 目錄底下，一定能找到編譯器所找不到的 .properties 檔，請將這個 .properties 檔複製到編譯器期望該檔案所在的目錄底下即可解決。
- 找不到 .class 檔:
在編譯 Java2 SDK 原始碼初期的時候常常出現找不到 .class 檔的錯誤訊息，筆者發現這些找不到的類別檔都是放置在 jdk1.3-src\ext\i18n\src\share\sun\io\底下，之所以會找不到這些.class 檔的緣故是因為根據 Makefile 裡頭的內容，jdk1.3-src\ext 目錄裡的原始碼會較晚編譯，而我們又關掉了 CLASSPATH 環境變數，導致在需要這些檔案的時候，這些檔案尚未從 .java 檔編譯成 .class 檔，很自然地編譯器就無法編譯。要解決這個問題，請讀者們先行到 jdk1.3-src\ext\i18n\src\share\sun\io\底下，輸入 javac *.java 以產生所有需要的 .class 檔，然後將這些新產生的.class 檔複製到 jdk1.3-src\build\win32\classes\sun\io 這個目錄底下。這樣一來，就再也不會遇到找不到 .class 檔的錯誤訊息了。
- Java2 SDK 原始碼的內附文件中要我們設定環境變數 TMP，但是實際的測試結果，發現在這個路徑似乎在編譯過程中一點用處也沒有，反倒是編譯的時候有用到環境變數 TEMP 所指向的目錄(在 Windows 2000 底下這個環境變數被設定為 %USERPROFILE%\Local Settings\Temp)，不知道是不是寫說明檔的人不小心漏打字。

當編譯工作告一段落之後，您將可以在 `jdk1.3-src\build\win32` 目錄下找到所有編譯過的執行檔以及 `.class` 檔，還有一些編譯過程中產生的中間檔。

`jdk1.3-src\build\win32\bin` 目錄下，您將會看到所有讓 Java2 SDK 正常運作的執行檔以及動態連結函式庫。又由於我們是使用 **make world** 指令，如果您觀察一下這個目錄，您會發現每個檔案都有最佳化版本與除錯版本(ex: `java.exe` 與 `java_g.exe`，其他以此類推)。

`jdk1.3-src\build\win32\lib` 目錄下將放置所有讓 Java 程式能夠順利執行的套件函式庫，例如 `i18n.jar` 等，在 `jdk1.3-src\build\win32\lib\ext` 目錄下可以找到 `iimp.jar` 等擴充套件還有一些資源檔。

`jdk1.3-src\build\win32\classes` 目錄下放置了最後會壓縮成 `rt.jar` 的所有 `.class` 檔(類別檔)。

另外，除了輸入

nmake world

指令之外，Makefile 之中定義了其他非 **world** 的編譯選項，請參閱下表：

編譯指令	作用
<code>nmake all</code>	建立 Java SDK 核心程式的除錯版本以及最佳化版本
<code>nmake ext</code>	建立 Java SDK 擴充程式的除錯版本以及最佳化版本
<code>nmake world</code> (同 <code>nmake world-all</code>)	建立 all 與 ext 中所有的程式
<code>nmake releasedocs</code>	執行 <code>javadoc.exe</code> 以產生 Java SDK 說明文件
<code>nmake alldocs</code>	執行 <code>javadoc.exe</code> 以產生在 <code>sun.*</code> 套件之下程式的所有說明文件
<code>Nmake world-clobber</code>	刪除前一次編譯之後產生的所有檔案和目錄

如果您想要了解所有的選項，請您自行開啟 `jdk1.3-src\build\win32\Makefile` 或是 `jdk1.3-src\build\win32\makefiles\defs.nmk` 這兩個檔案做更深入的研究。

■ 接下來

有關 Java 2 SDK 原始碼的概觀介紹就到這裡，相信大家都已經有了堅實的基礎，如果您已經是一位非常有經驗的工程師，相信您已經能夠隨意地修改出自己版本的 Java2 SDK 了！

■ 網路資源

名稱	URL
Java Developer Connection	http://developer.java.sun.com/developer/
Java2 原始碼下載	http://www.sun.com/software/communitysource/java2/