CHAPTER B

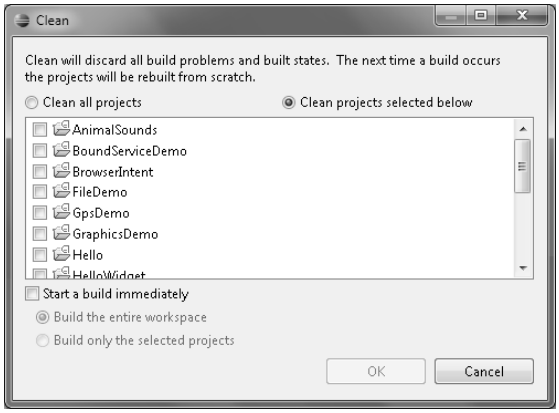
ปัญหาที่พบบ่อย และวิธีแก้ไข

ภาคผนวกนี้รวบรวมปัญหาที่พบได้บ่อยในการพัฒนาแอพแอนดรอยด์ พร้อมทั้งวิธีแก้ไข

การคลีน (Clean) โปรเจ็ค

ในขณะที่พัฒนาแอพแอนดรอยด์ด้วย Eclipse ถ้าคุณเจอปัญหาแปลกๆที่ไม่รู้ว่าจะแก้ไขอย่างไร วิธีหนึ่งที่อาจช่วยได้ก็คือการสั่งคลีน (clean) โปรเจ็ค ซึ่งหมายถึงการลบไฟล์ผลลัพธ์ที่ได้จากการบิวด์ (build) โปรเจ็ค เพื่อให้ Eclipse บิวด์โปรเจ็คและสร้างไฟล์ผลลัพธ์ขึ้นมาใหม่

การคลีนโปรเจ็คจะทำได้โดยคลิกเมนู **Project ▸ Clean** จะปรากฏวินโดว์ดังรูป คุณสามารถเลือกคลีนทุกโปรเจ็คในเวิร์คสเปซ (ออปชั่น **Clean all projects**) หรือเฉพาะโปรเจ็คที่ต้องการ (ออปชั่น **Clean projects selected below**) นอกจากนี้ยังสามารถกำหนดให้ Eclipse บิวด์โปรเจ็คใหม่ทันที หลังจากคลีนแล้ว (ออปชั่น **Start a build immediately**)



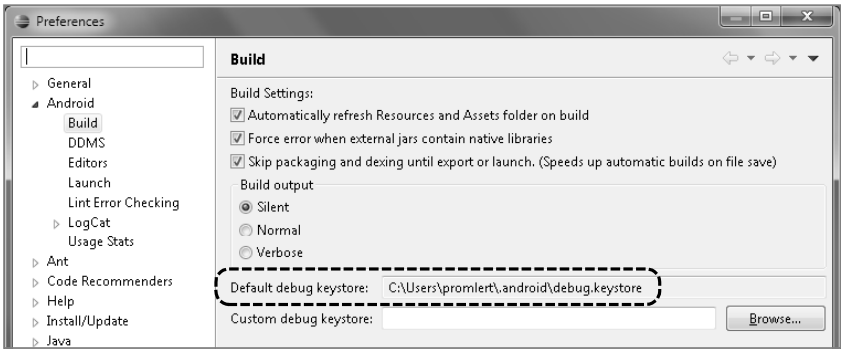
มีรายงานว่าการคลีนโปรเจ็คสามารถแก้ปัญหาเหล่านี้ได้

- Project *xxx* is missing required source folder: 'gen'
- The project could not be built until build path errors are resolved.
- Unable to open class file R.java.

ปัญหา Debug Certificate (Debug Key) หมดอายุ

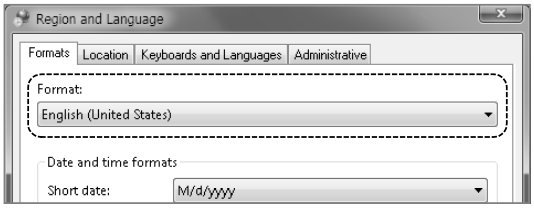
เครื่องพีซีที่ตั้งค่าให้แสดงวันที่เป็นภาษาไทย (ปี พ.ศ.) จะพบปัญหา Debug Certificate หมดอายุเมื่อสร้างโปรเจ็คใหม่หรือสั่งรันโปรเจ็ค ให้แก้ไขดังนี้

1 ตรวจสอบว่า Eclipse เก็บไฟล์ debug.keystore ไว้ที่ใด โดยคลิกเมนู **Window ▸ Preferences** แล้วคลิกหัวข้อ **Android ▸ Build** ทางซ้าย ชื่อพาธเต็มของไฟล์ debug.keystore จะแสดงอยู่ที่ช่อง Default debug keystore



2 เมื่อรู้ตำแหน่งของไฟล์ debug.keystore แล้ว ให้ลบไฟล์นั้น

3 ไปที่ Region and Language (หรือ Regional and Language Options) ใน Control Panel แล้วเปลี่ยนรูปแบบวันเวลาให้เป็นแบบ **English (United States)** แล้วคลิก **Apply**



4 กลับไปที่ Eclipse แล้วสั่งคลีนโปรเจ็คทั้งหมด (ดูหัวข้อ “การคลีนโปรเจ็ค” ก่อนหน้านี้) ซึ่งจะทำให้มีการสร้างไฟล์ debug.keystore ขึ้นมาใหม่

ไฟล์ debug.keystore จะถูกสร้างขึ้นเพียงครั้งเดียว หลังจากนั้นไม่ว่าคุณจะสร้างโปรเจ็คใหม่อีกกี่โปรเจ็ค ADT ก็จะใช้ข้อมูลจากไฟล์เดิมนี้ในการ sign แอพ ดังนั้นเมื่อได้ไฟล์ debug.keystore ที่ใช้งานได้แล้ว คุณก็สามารถเปลี่ยนวันที่กลับมาเป็นภาษาไทย โดยไม่ต้องกลัวว่าจะเกิดปัญหา Debug Certificate หมดอายุอีก

5 ไปที่ Region and Language (หรือ Regional and Language Options) ใน Control Panel แล้วเปลี่ยนรูปแบบวันเวลากลับมาเป็น **Thai (Thailand)**

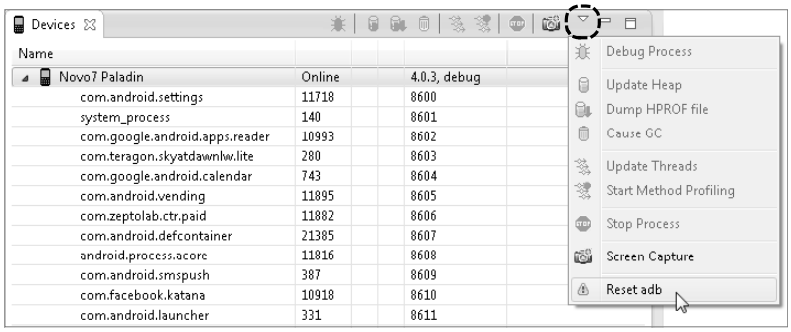
ปัญหาของ Android Debug Bridge (ADB)

Android Debug Bridge หรือคำสั่ง adb ใน Android SDK (C:\android-sdk\platform-tools\adb.exe) คือโปรแกรมที่ช่วยให้เราติดต่อกับอีมูเลเตอร์ หรืออุปกรณ์แอนดรอยด์ได้ โดยการพิมพ์คำสั่งทาง command line

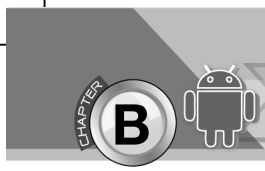
โดยปกติเราไม่ได้เรียกใช้ adb เอง แต่ Eclipse จะเรียกให้อัตโนมัติ โดยการทำงานของ Eclipse ที่ต้องติดต่อกับอีมูเลเตอร์ หรืออุปกรณ์แอนดรอยด์ เช่น การส่งแอพไปติดตั้ง หรือการดีบักแอพ โดยรันโค้ดทีละบรรทัดนั้น Eclipse จะไปเรียกใช้ adb อีกทีหนึ่ง

บางครั้งการทำงานของ adb อาจมีปัญหา ทำให้ Eclipse ติดต่ออีมูเลเตอร์หรืออุปกรณ์แอนดรอยด์ไม่ได้ ซึ่งจะแก้ไขได้โดยรีสตาร์ท adb ดังนี้

1 คลิกเมนู **Window ▸ Open Perspective ▸ DDMS** เพื่อเปิด DDMS Perspective



2 ที่วินโดว์ Devices (Devices View) ให้คลิก **View Menu** แล้วคลิก **Reset adb**



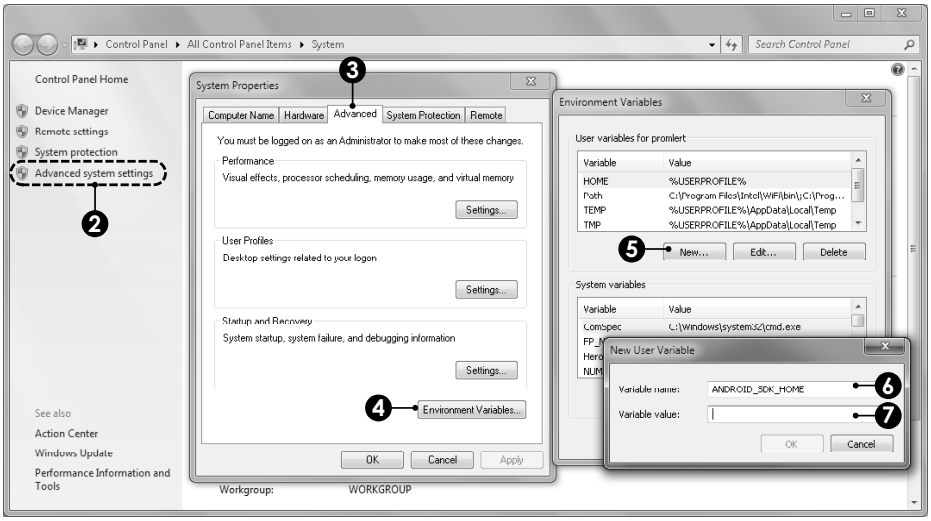
ปัญหา: หารีซอร์สที่มีการอ้างอิงในไฟล์ R.java ไม่พบ

ถ้าหาก Eclipse แจ้งว่าไม่พบรีซอร์สที่มีการอ้างอิงอยู่ในไฟล์ R.java เช่น `R.layout.main` ให้คุณตรวจสอบให้แน่ใจว่าไม่ได้มีการอิมพอร์ต `android.R` เข้ามาในโค้ดจาวา เพราะการอิมพอร์ต `android.R` จะทำให้ Eclipse ไม่สนใจไฟล์ R.java และนำมาสู่ปัญหาดังกล่าว

ปัญหา: รันอีมูเลเตอร์ไม่ได้

กรณีที่รันอีมูเลเตอร์ไม่ได้นั้นส่วนใหญ่มีสาเหตุมาจากชื่อพาธที่เก็บไฟล์ .avd มีช่องว่างหรือไม่ก็ตัวอักษรภาษาไทยรวมอยู่ วิธีแก้ปัญหาคือ ให้คุณเปลี่ยนตำแหน่งสำหรับเก็บไฟล์ .avd โดยสร้างตัวแปร environment ชื่อ ANDROID_SDK_HOME ให้ชี้ไปยังตำแหน่งใหม่ ตามขั้นตอนดังนี้

1 คลิกขวาที่ My Computer แล้วเลือก Properties



2 คลิก **Advanced system settings** ทางซ้าย

3 คลิกแท็บ **Advanced**

4 คลิกปุ่ม **Environment Variables**

5 คลิกปุ่ม **New** ในส่วนของ User variables ข้างบน

6 ป้อน Variable name ว่า **ANDROID_SDK_HOME**

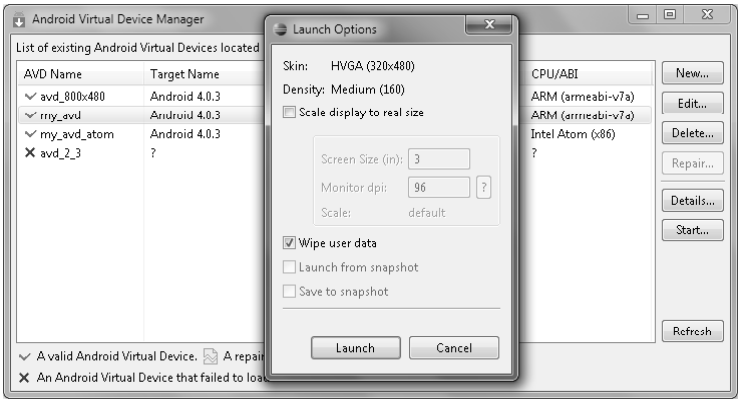
7 ป้อน Variable value เป็นชื่อไดเรคทอรีที่จะใช้เก็บไฟล์ .avd (คุณต้องสร้างไดเรคทอรีนี้ขึ้นมาด้วย)

8 รีสตาร์ท Eclipse

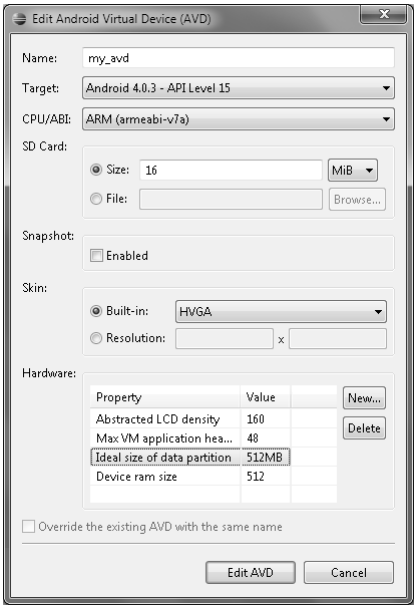
ปัญหา: ติดตั้งแอพไม่ได้ เนื่องจากเนื้อที่เก็บข้อมูลไม่พอ

บางครั้งเมื่อคุณรันโปรเจ็ค อาจเกิดข้อผิดพลาดว่าไม่สามารถติดตั้งแอพลงในอีมูเลเตอร์ได้เพราะเนื้อที่เก็บข้อมูลไม่พอ (INSTALL_FAILED_INSUFFICIENT_STORAGE) เนื่องจากอีมูเลเตอร์เตรียมเนื้อที่ไว้สำหรับติดตั้งแอพเพียง 64 MB เท่านั้น

คุณอาจแก้ปัญหาด้วยการลบแอพที่ติดตั้งไว้ในอีมูเลเตอร์ทิ้งไป โดยเช็คเลือก **Wipe user data** ใน Launch Options ตอนที่สั่งรันอีมูเลเตอร์



หรืออีกวิธีคือให้แก้ไขคุณสมบัติของอีมูเลเตอร์ แล้วใช้ปุ่ม **New** เพิ่มหัวข้อ **Ideal size of data partition** และกำหนดขนาดให้มากกว่า 64 MB



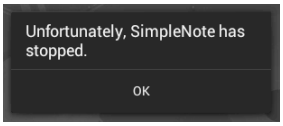
ปัญหา: ไม่รู้จักชนิดข้อมูล

เมื่อคุณประกาศตัวแปรหรือพารามิเตอร์ของเมธอด แล้ว Eclipse ขีดเส้นใต้สีแดงที่ชื่อชนิดข้อมูลและแจ้งข้อผิดพลาดว่า "*XX* cannot be resolved to a variable" แสดงว่าคุณยังไม่ได้อิมพอร์ตชนิดข้อมูล (คลาส) นั้นเข้ามา

วิธีแก้ไขคือ ให้คลิกขวาตรงไหนก็ได้ใน Code Editor แล้วเลือก **Source ▸ Organize Imports** เพื่อให้ Eclipse เพิ่มการอิมพอร์ตทั้งหมดที่จำเป็นให้อัตโนมัติ

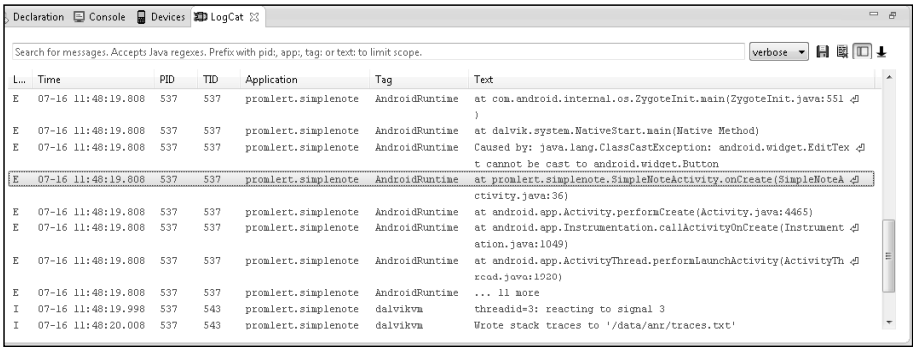
ปัญหา: Force close และ App has stopped

ปัญหา Force close (บังคับปิด) และ App has stopped (แอพหยุดทำงาน) โดยทั่วไปเกิดจากมีข้อผิดพลาดเกิดขึ้นในช่วงรันแอพ (Runtime Error หรือ Exception) ซึ่งเราสามารถรู้ตำแหน่งในโค้ดที่ทำให้เกิดข้อผิดพลาดได้โดยใช้วินโดว์ LogCat (LogCat View) ของ Eclipse



การเปิดวินโดว์ LogCat ให้คลิกเมนู **Window ▸ Show View ▸ Other ▸ Android ▸ LogCat**

วินโดว์ LogCat จะแสดงข้อความ log (log messages) ที่แอนดรอยด์บันทึกเก็บไว้ในอีมูเลเตอร์/อุปกรณ์ ซึ่งในกรณีปกติข้อความเหล่านี้จะเป็นข้อมูลที่บอกสถานะหรือคำเตือนต่างๆที่ไม่ร้ายแรง แต่เมื่อแอพของคุณทำงานผิดพลาด LogCat จะแสดงข้อความแจ้งข้อผิดพลาดเป็นสีแดงและมีตัวอักษร E อยู่ในคอลัมน์ซ้ายมือสุด ดังรูป



ข้อความแจ้งข้อผิดพลาดมักจะมาติดๆกันหลายข้อความ ซึ่งบอกตำแหน่งที่เกิดข้อผิดพลาดในแต่ละเมธอดที่มีการเรียกใช้ต่อๆกันมา (call chain) แน่นอนว่าจุดที่เราจะสนใจก็คือตำแหน่งในโค้ดของเรา ให้คุณไล่ดูในคอลัมน์ Text จนกระทั่งเจอชื่อเนมสเปซของแอพของคุณ (เนมสเปซที่กำหนดตอนสร้างโปรเจ็ค) ในที่นี้คือรายการที่เขียนว่า

```
at promlert.simplenote.SimpleNoteActivity.onCreate(SimepleNoteActivity.java:36)
```

เมื่อเจอแล้ว ให้ดูในวงเล็บคุณก็จะรู้ว่าข้อผิดพลาดนั้นเกิดที่ไฟล์ใดและที่บรรทัดใด

สำหรับสาเหตุหรือชนิดข้อผิดพลาด ให้ดูย้อนขึ้นไปจนเจอบรรทัดที่ระบุถึงชนิด Exception ซึ่งในที่นี้คือ

```
Caused by: java.lang.ClassCastException: android.widget.EditText cannot be
cast to android.widget.Button
```

นั่นคือเกิดข้อผิดพลาดชนิด `ClassCastException` และมีคำอธิบายเพิ่มเติมว่า ไม่สามารถแปลง (cast) `EditText` ไปเป็น `Button` ได้

การดีบักโค้ด

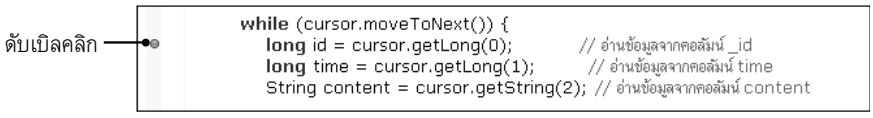
บางครั้งสาเหตุของข้อผิดพลาดที่แจ้งในวินโดว์ LogCat ก็อาจไม่ได้ช่วยให้คุณเข้าใจว่าปัญหาเกิดได้อย่างไร และควรแก้ไขอย่างไร ดังนั้นเพื่อที่จะค้นหาสาเหตุของข้อผิดพลาด สิ่งที่คุณต้องทำต่อไปก็คือการดีบักโค้ด ซึ่งทำได้ 2 รูปแบบ คือการพิมพ์ค่าหรือสถานะต่างๆในโค้ด (เช่นค่าของตัวแปร) ออกมาใน LogCat เอง และการใช้เครื่องมือดีบักของ Eclipse เพื่อไล่ดูการทำงานของโค้ดทีละบรรทัด

การพิมพ์ค่าออกมาใน LogCat

ทำได้โดยอิมพอร์ตคลาส `android.util.Log` เข้ามาในโค้ดจาวา แล้วเรียกใช้เมธอด `d` ของคลาสดังกล่าว ตัวอย่างการใช้งานขอให้ดูในบทที่ 6 หัวข้อ “การเริ่มเกมใหม่” ซึ่งเราจะพิมพ์ข้อความดีบักลงใน LogCat เพื่อแสดงว่าผู้ใช้เลือกระดับความยากเป็นตัวเลือกใดตอนเริ่มเกมใหม่

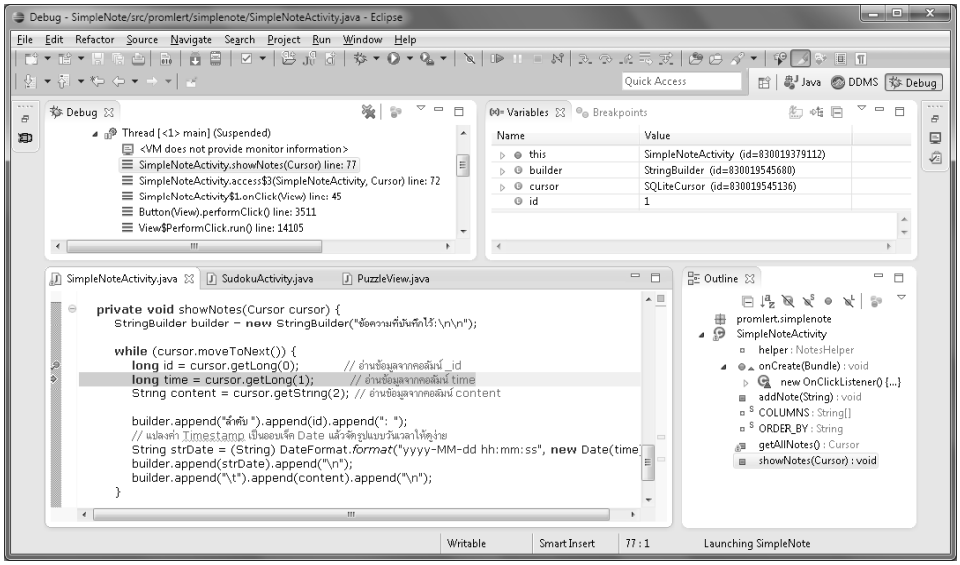
การใช้เครื่องมือดีบักของ Eclipse

เริ่มจากให้คุณกำหนดเบรคพอยต์ (Breakpoint) ในโค้ดตรงบรรทัดที่คุณต้องการให้แอพหยุดการทำงานชั่วคราว โดยดับเบิลคลิกที่แถบสีเทาข้างหน้าบรรทัดนั้นให้ปรากฏจุดกลมสีฟ้า ดังรูป



จากนั้นให้สั่งดีบักโดยคลิกเมนู **Run ▸ Debug** หรือกดคีย์ลัด F11 เมื่อแอพทำงานจนถึงตำแหน่งที่คุณกำหนดเบรคพอยต์ไว้ Eclipse ก็จะถามว่าต้องการเปิด Debug Perspective หรือไม่ ให้ตอบ Yes

เมื่อเข้าสู่ Debug Perspective แล้ว Eclipse จะไฮไลท์บรรทัดที่เรากำหนดเบรคพอยต์ บอกให้รู้ว่าบรรทัดนี้จะถูกประมวลผลเป็นลำดับถัดไป คุณสามารถใช้คำสั่ง Step Into (คีย์ลัด F5), Step Over (คีย์ลัด F6) หรือ Step Return (คีย์ลัด F7) จากเมนู Run เพื่อรันโค้ดไปทีละบรรทัด โดยความแตกต่างของสามคำสั่งนี้คือ กรณีเป็นการเรียกเมธอด คำสั่ง Step Over จะรันเมธอดจนจบและข้ามไปยังโค้ดในบรรทัดถัดไปเลย ส่วน Step Into จะรันเข้าไปในเมธอด เพื่อให้คุณไล่ดูการทำงานภายในเมธอดได้ ส่วน Step Return จะใช้กลับออกมาจากเมธอดหากใช้คำสั่ง Step Into เข้าไป



ในขณะที่รันโค้ดทีละบรรทัด คุณสามารถตรวจสอบค่าของตัวแปรต่างๆได้จากวินโดว์ Variables ที่มุมบนขวาของ Eclipse (ในรูปนี้ผู้เขียนใช้คำสั่ง Step Over ไปแล้ว 1 ครั้ง)

NOTE

- ข้อมูลข่าวสารเกี่ยวกับหนังสือ www.facebook.com/AndroidDevBook
- พูดคุย สอบถาม แนะนำเกี่ยวกับเนื้อหาของหนังสือ www.facebook.com/groups/AndroidDevBook/