Eclipse 的 Debug 介绍与技巧

没有任何程序员能够一气呵成的写出没有任何 Bug 的代码，所以很多程序员有相当一部分时间是花费在 Debug 上的，程序调试是每个程序员必须面对的工作。如何使用 Eclipse 进行有效的、尤其是高效地调试代码是一个值得学习的技巧。

[一、Eclipse调试介绍](http://blog.jobbole.com/93421/#Eclipse调试介绍)

[二、Eclipse中和Debug相关的视图](http://blog.jobbole.com/93421/#Eclipse中和Debug相关的视图)

* 2.1 Debug View
* 2.2 Variables View
* 2.3 Breakpoints View
* 2.4 Expressions View
* 2.5 Display View

[三、Debug](http://blog.jobbole.com/93421/#Debug)

[3.1 设置断点](http://blog.jobbole.com/93421/#设置断点)

[3.2 调试程序](http://blog.jobbole.com/93421/#调试程序)

3.2.1 调试本地 Java 语言程序

3.3.2 远程调试

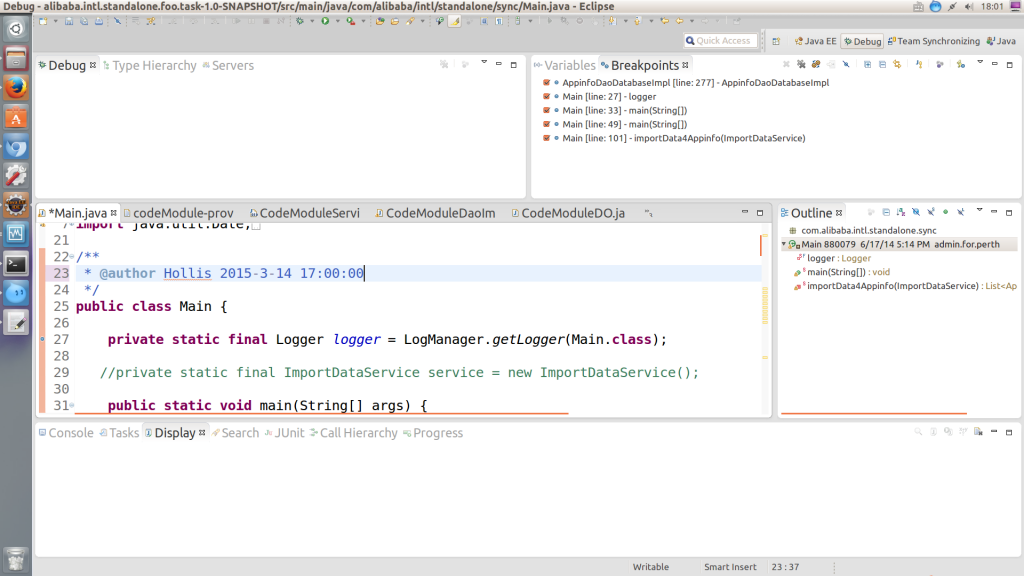
一、Eclipse调试介绍

使用过Eclipse的人都知道，Eclipse都是自带Java调试器的，可以提供很多基本的调试功能。 Eclipse 平台工作台及其工具是围绕 JDT 组件构建的，该组件为 Eclipse 提供了下列特性：

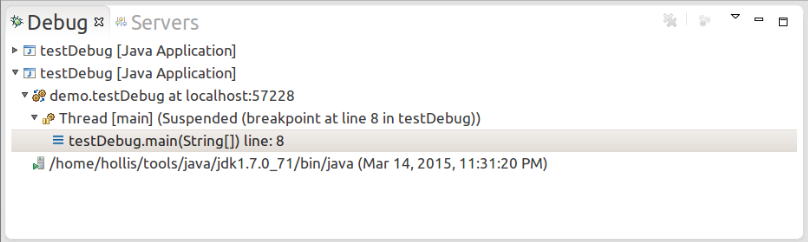
项目管理工具 透视图和视图 构造器、编辑器、搜索和构建功能 调试器

**二、Eclipse中和Debug相关的视图**

**图 1. Eclipse Debug 透视图的一般视图（该视图可以在Eclipse界面的右上角位置点击Debug按钮进入）**

[](http://www.hollischuang.com/wp-content/uploads/2015/05/DebugView.png)

Debug View:

[](http://www.hollischuang.com/wp-content/uploads/2015/05/debug.png)

Debug视图允许您在工作台上管理正在调试和运行的程序，他显示了你正在调试的程序中挂起的线程的堆栈帧，程序中的每个线程作为树的节点出现。他展示了正在运行的每个目标的进程。如果线程被挂起，它的堆栈帧以子元素的形式展示。以下是一些常用的debug按钮：

[skip_brkp](http://www.hollischuang.com/wp-content/uploads/2015/05/skip_brkp.png)

**Skip All Breakpoints:**:将所有断点设置为被跳过的，设置了Skip All Breakpoints之后，所有断点上都会有一个斜线，表示断点将被跳过，线程不会在该断点处被挂起。

[drop_to_frame](http://www.hollischuang.com/wp-content/uploads/2015/05/drop_to_frame.png)

**Drop to Frame：**这个命令可以让程序回到当前方法的开头第一行重新开始执行，可以重新执行这个java堆栈帧，可以选择一个指定的堆栈帧，然后点击 Drop to Frame，这样就可以重新进入指定的堆栈帧。使用Drop to Frame时候需要注意：

1.不能drop到已经执行过的方法栈中的方法中。

2.drop到stack frame中时，不会改变全局数据原有的值，比如，一个包含元素的vertor并不会被清空。

[stepbystep_co](http://www.hollischuang.com/wp-content/uploads/2015/05/stepbystep_co.png)

**Step Filters:**这个功能比较简单，就是当我们在debug的时候想要忽略一些我们不关注的类时，可以开启Step Filters进行过滤，程序会一直执行直到遇到未经过滤的位置或断点。Step Filters功能由Use Step Filters，Edit Step Filters，Filter Type，Filter Package四项组成。具体操作如下：

步骤 1： Windows -> Preferences -> Java -> Debug -> Step Filtering.

步骤 2：选择‘Use Step Filters’.

步骤 3：在屏幕上选中所需的选项。你可以添加你自己代码库中的部分代码。

步骤 4：点击‘Apply’.

原理上，Edit Step Filter命令用于配置Step Filter规则，而Filter Type与Filter Package分别指的是过滤的Java类型与Java Package。

[stepreturn_co](http://www.hollischuang.com/wp-content/uploads/2015/05/stepreturn_co.png)

**Step Return:**跳出当前方法，在被调用方法的执行过程中，使用Step Return会在执行完当前方法的全部代码后跳出该方法返回到调用该方法的方法中。

[stepover_co](http://www.hollischuang.com/wp-content/uploads/2015/05/stepover_co.png)

**Step Over:**在单步执行时，在函数内遇到子函数时不会进入子函数内单步执行，而是将子函数整个执行完在停止，也就是把子函数整个作为一步。

[stepinto_co](http://www.hollischuang.com/wp-content/uploads/2015/05/stepinto_co.png)

**Step Into：**单步执行，遇到子函数就进入并且继续单步执行

[resume_co](http://www.hollischuang.com/wp-content/uploads/2015/05/resume_co.png)

**Resume：**恢复暂停的线程，直接从当前位置跳到下一个断点位置。

[suspend_co](http://www.hollischuang.com/wp-content/uploads/2015/05/suspend_co.png)

**Suspend：**暂停选定的线程，这个时候可以进行浏览或者修改代码，检查数据等。

Eclipse通过Suspend与Resume来支持线程的暂挂与恢复。一般来讲，Suspend适用于多线程程序的调试，当需要查看某一个线程的堆栈帧及变量值时，我们可以通过Suspend命令将该线程暂挂。Resume用于恢复。有两种Resume需要注意：第一是当在调试过程中修改程序代码，然后保存，点击Resume，此时程序会暂挂于断点。第二是当程序抛出异常时，运行Resume，程序也会暂挂于断点。

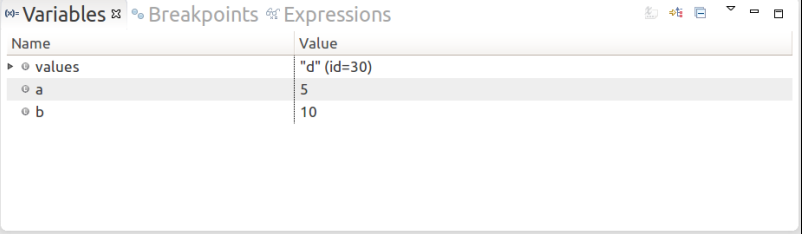
[terminate_co](http://www.hollischuang.com/wp-content/uploads/2015/05/terminate_co.png)

**Terminate:**Eclipse通过Terminate命令终止对本地程序的调试。

[disconnect_co](http://www.hollischuang.com/wp-content/uploads/2015/05/disconnect_co.png)

**Disconnect:**Eclipse使用Disconnect命令来终止与远程JVM的socket连接。

**Variables View ：**

[](http://www.hollischuang.com/wp-content/uploads/2015/05/velue.png)

Variables View显示与Debug View中选定的堆栈帧相关的变量信息，调试Java程序时，变量可以选择将更详细的信息显示在详细信息窗格中。此外，Java对象还可以显示出其包含的属性的值。在该窗口中选中变量鼠标右键点击可以进行许多操作，主要操作有以下这些：

**All Instances：**打开一个对话框来显示该java类的所有实例，使用该功能需要java虚拟机支持实例的检索。

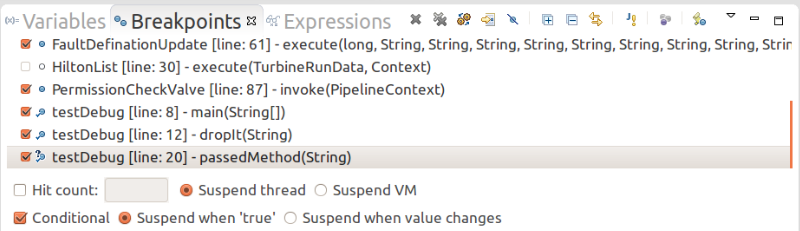
**All References:**:打开一个对话框来显示所有引用了该变量的java对象，

**Change Value：**：更改变量的值，该功能可以和Drop to Frame联合使用进行程序的调试。使用这两个功能就可以代替重新debug

**Copy Variables：**复制变量的值，尤其在变量值很长（比如json数据）的时候，这个功能就派上用场了。

**Find：**有的时候一个类中变量特别多的时候，可以进行查找。

**Breakpoints View ：**

[](http://www.hollischuang.com/wp-content/uploads/2015/05/break.png)

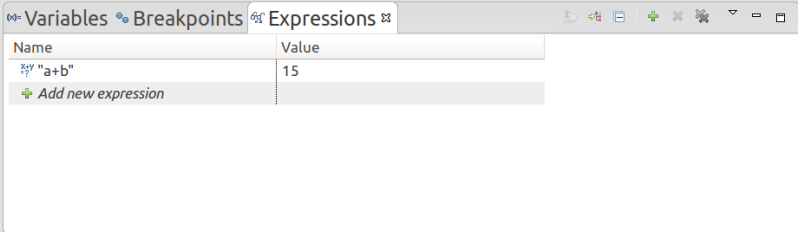
Breakpoints View将列出你在当前工作区间里设置的所有断点，双击断点可以进入到程序中该断点的位置。还可以启用或禁用断点，删除，添加新的，根据工作组或点命中计数给他们分组。在使用断点是有以下两个技巧是十分有用的：

**Hit Count:** 是指定断点处的代码段运行多少次，最典型的就是循环，如果要让一个循环执行10次就线程挂起，则指定Hit Count值为10，那么当前的循环执行到第九次的时候就会挂掉。

**Conditional:**顾名思义，就是条件判断，例如我们需要循环变量i==10时，线程挂起，则条件设定为i==10,选择Suspend when “true”。

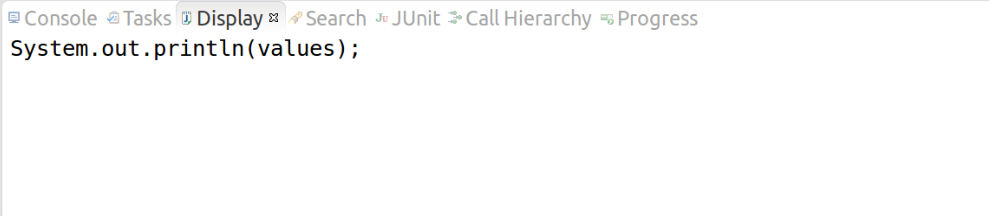
那如果上面的Hit Count和Conditional都选择的话，如果表达式和值设置不合理则会失效。如果选择Suspend when value changes，那么可能在Conditional在变量值发生改变的时候就挂起。

**Expressions View ：**

[](http://www.hollischuang.com/wp-content/uploads/2015/05/express.png)

要在 Debug 透视图的编辑器中求表达式的值，选中设置有断点的一整行，并在上下文菜单中选择 Inspect 选项。表达式是在当前堆栈帧的上下文中求值的，其结果显示在 Display 窗口的 Expressions 视图中。 比如我想要计算变量a+b的值，那么就可以在表达式视图中加一个表达式：a+b

**Display View ：**

[](http://www.hollischuang.com/wp-content/uploads/2015/05/disp.png)

可以使用这个视图，输入或者演算一些新的代码。这些代码在当前的调试位置的上下文环境中被执行，这意味着，你可以使用所有变量甚至是内容助手。要执行你的代码的话，只需标记它，并使用右键菜单或者CTRL+U(执行)或者 CTRL+SHIFT+I (检查).

**三、Debug**

**设置断点**

在源代码文件中，在想要设置断点的代码行的前面的标记行处，双击鼠标左键就可以设置断点，在相同位置再次双击即可取消断点。有的时候我们还有这样的需要，就是我并不想一行一行的执行代码，比如一个for循环会循环1000多遍，我只想在第500遍的时候让线程挂起进行调试，这个时候我们可以使用**条件断点**。 **设置条件断点：我们可以给该断点设置触发条件，一旦满足某条件是才开始调试，可以在断点处点击鼠标右键，选择Breakpoint Properties进入断点设置页面，**刚刚在讲断点视图的时候我们学到过Hit Count和Conditional的用法，这里可以设置条件和执行次数。

**调试程序**

**1、调试本地 Java 语言程序**

在所有调试中，调试一个Java程序是最简单的，主要有**设置断点**、**启动调试**、**单步执行**、**结束调试**几步。

**设置断点：**前面已经说过。

**启动调试：**Eclipse提供四种方式来启动程序(Launch)的调试，分别是通过菜单(Run –> Debug)、图标(“绿色臭虫”)、右键->Debug As以及快捷键(F11)，在这一点上，与其他命令(例如Run)类似。

**单步执行：**主要使用前面讲过的几个视图进行调试，其中debug视图中的几个按钮有快捷键:

Step Retuen(F7)

Step Over (F6)

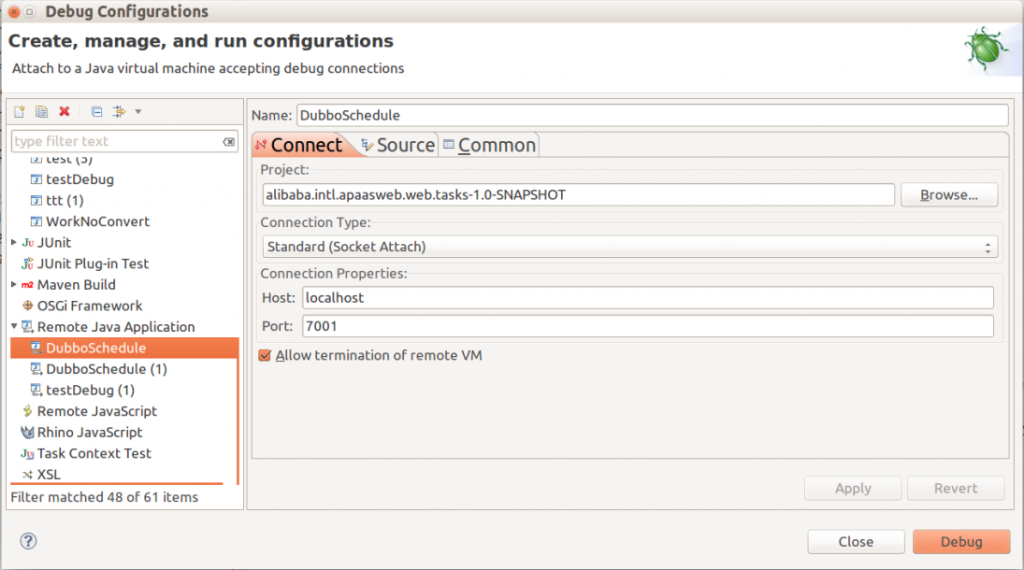
Step Into (F5)

**结束调试：**通过Terminate命令终止对本地程序的调试。

**2、远程调试**

远程调试主要用于调试非本地Java程序，这里的非本地并不是只在其他人的机器上才叫非本地。运行在本机上的Web服务器上的程序在调试时也需要使用远程调试。远程调试的大致步骤和调试本地Java语言程序基本一直，只是在设置上有些不同。

Eclipse 调试器可以调试远程应用程序。它可以连接到一个运行 Java 应用程序的远程 VM，将自己连接到该应用程序上去。使用远程调试会话与使用本地调试会话大致相同。但是，远程调试配置需要在 Run > Debug 窗口中配置一些不同的设置。需要在左侧视图中先选择 Remote Java Application 选项，然后单击 New。这样就创建了一个新的远程启动配置，会显示出三个选项卡：Connect、Source 和 Common。

[](http://www.hollischuang.com/wp-content/uploads/2015/05/QQ%E6%88%AA%E5%9B%BE20150314215010.png)

在 Connect 选项卡的 Project 字段，选择在启动搜索源代码时要引用的项目。在 Connect 选项卡的 Host 字段，输入运行 Java 程序的远程主机的 IP 地址或域名。在 Connect 选项卡的 Port 字段，输入远程 VM 接收连接的端口。通常，该端口在启动远程 VM 时指定。如果想让调试器决定在远程会话中 Terminate 命令是否可用，可以选择 Allow termination of remote VM 选项。如果希望可以终止连接的 VM，则选择该选项。现在，在选择 Debug 选项时，调试器会尝试连接到指定地址或端口的远程 VM，结果会在 Debug 视图中显示出来。