

Eclipse的调试功能的10个小窍门 - 歌颂者 - 博客园

本文由 [ImportNew](#) - [唐小姐](#) 翻译自 [Codecentric](#)。如需转载本文，请先参见文章末尾处的转载要求。

你可以已经看过一些类似“关于调试的N件事”的文章了。但我想我每天大概在调试上会花掉1个小时，这是非常多的时间了。所以非常值得我们来了解一些用得到的功能，可以帮我们节约很多时间。所以在这个主题上值得我再来写一篇文章。

第一条：不要过分的调试！

有关调试的第一条要牢记的便是这条很疯狂的口号！但是我必须要在这一再说一遍：不要过分的调试！试着将复杂的逻辑分解成独立的小单元，然后写单元测试代码来保证小单元的正确运行。我经常看到某些人会这么做：在一个大型的Web应用上点击，填了几个表单，跳转了多个页面，只是为了确认最后一个页面的结果的正确性，最后在调试视图下来开发代码。

在你开启tomcat之前，应该要先问问自己：有没有什么方法可以用单元测试来检测代码的行为呢？你可以找到很多教你如何写出好的代码的教程。而这里我主要来谈一谈Eclipse的调试功能，你可能不知道，或者长时间以来有些淡忘的功能。

调试视图：有条件的断点

如果你仅仅对程序的某个部分感兴趣，调试视图是非常有帮助的。假设你想看看一个循环中的第十三次循环得到什么结果，或者你正在调试一个抽象父类，想看看某个具体的子类。你可以在调试视图设置条件，右键点击代码旁的蓝色断点符号，在弹出菜单中选择“Breakpoint Properties...”。你可以选择是在你的代码段返回真值是停止程序或者当你的代码段的值改变时停止。

变量视图：显示逻辑结构(Show Logical Structure)

如果你想在变量视图中查看Map或者List中的值，eclipse的默认设置可能不是那么方便。譬如你在使用HashMap，你必须要点开所有的子节点才能看到HashMap中的内容，还要受到HashMap实现细节的干扰。但是在变量上有一个小按钮-“Show Logical Structure”。它非常的方便，尤其当你没有实现某个对象的toString()代码时。我的老板前几个星期刚刚告诉我Eclipse有这个功能。你知道，他是那种每天只用对着PowerPoint和Excel的人。这对我这种号称程序员的家伙来说是件多么丢脸的事情啊！;-)

变量视图：更改值...(Change Value...)

假设你在一个基于Web的表单中稍微改了一点输入值，那么我们不需要重启调试session，你可以直接使用变量视图的改变变量值的功能。这可以节省你的时间，有时候也能帮你模拟一些奇怪的行为。

显示视图(Display View)

你知道还有个“显示视图”吗？你可以通过“Window” -> “Show View” -> “Display”激活这个视图。现在你的Eclipse有一个完全空的视图。你可以通过这个视图来输入以及验证新的代码段。这些代码将会在当前的调试的位置的上下文中执行，这意味着你可以使用所有的变量，你甚至还能使用内容辅助。要执行这个代码，你需要选中它，然后点击右键弹出菜单点击相应的项，或者按下CTRL+U(执行)或者按下CTRL+SHIFT+I(检查)。

导航：Drop to Frame

我相信所有人都知道“Step Into”，“Step over”，甚至知道“Step return”。这是调试要掌握的最基本的技巧。我还想告诉你另外两种方法。我非常喜欢它们。第一个是“Drop to Frame”。有了这个功能，你可以回到过去;-)，你可以轻松回到你曾经运行过的Java stackframe中的位置。这对我来说非常有用，因为可能我错过了某一点，有了“Drop to Frame”功能，我可以轻松重新运行那段代码了。

导航：Step into Selection

第二个是“Step into Selection”。这个功能非常简单，但是很多人没有使用。你仅仅需要按下Ctrl+Alt，同时点击你想去的方法上点击，快捷方便。相比较“Step Into”，“Step into Selection”更加方便。譬如假设你想进入某个有许多参数的方法，你可以使用一步就跳过所有参数的赋值。”Run to line”也是个不错的功能。你只需要将鼠标放在那一行前面，然后点击“CTRL+R”。

导航：使用快捷键

如果你不再使用鼠标，你会变得更加有效率。至少你应该掌握以下的快捷键：

F5 - “Step Into”

F6 - “Step Over”
F7 - “Step Return”
F8 - “Resume”
Ctrl+Shift+B - “Toggle Breakpoint”
Ctrl+Shift+I - “Inspect”

断点视图: Watchpoints

是什么改变了变量?!有时创建watchpoint会有很大的帮助。当某个要监视的field的值改变时或是读到时，调试器会停下来。双击某个field，这个watchpoint显示在断点视图中，你可以编辑它的属性。你也可以设置hit count，意思是当你输入的hit count已经达到了，调试器便会停下来。这个也适合于普通的断点。

清晰易读的对象

变量视图是使用toString()方法来显示一个对象的值，所以我们应该要写出清晰易读的toString()方法。

java.lang.Object默认的toString()方法的javadoc也如此建议:

	<pre>* Returns a string representation of the object. In general, the toString method returns a string that textually represents this object. The result should be a concise but informative representation that is easy for a person to read. * It is recommended that all subclasses override this method. [...]</pre>
1	
2	
3	
4	
5	
6	
7	

你可以看一看commons-lang中的[ToStringBuilder](#)。它可以帮助你写出“良好的、前后一致的”toString()方法。

如果你不能修改toString()方法，譬如你使用框架，或是外来的API，在Eclipse中创建“Detail Fomatter”可以帮到你。在变量视图中右击某个对象，然后点击“New Detail Formatter...”。然后你便能够定义代码，之后这种类型的Object都用这种形式来显示。