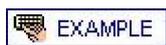


调试工具

引论

调试工具是一种能够发现所执行文本中的错误或失速问题的工具。调试工具的作用在于你能够监控每个单指令，使其必须在整个一连串指令内执行。通过这种方式你可以真正看到由一个单指令引起的变化，而不是看到许多指令完成后的变化。就像动画效果是模拟的形象化一样，调试工具也是执行的形象化。而且正如动画减慢了模拟的过程一样，调试工具减慢了所有的执行过程。这就是为什么启动Enterprise Dynamics时要关闭调试工具的原因。

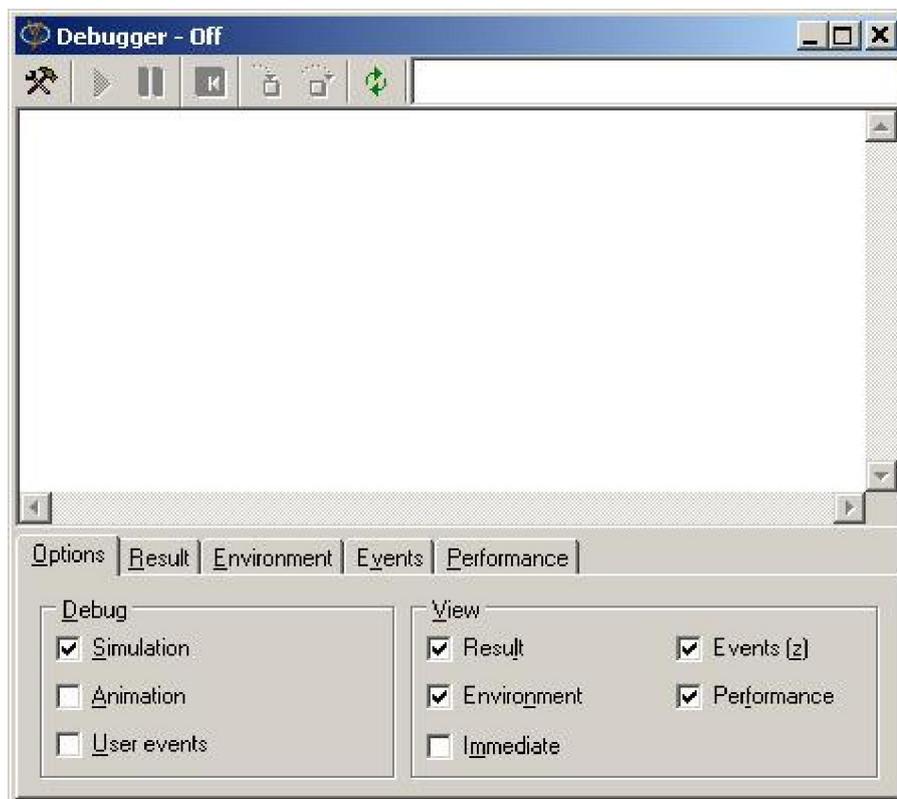
就像ED中的所有其他功能一样，调试工具也有自己的窗口。为了显示窗口，你可以使用4DScript图形化调试工具，它只有一个参数：显示（1）或者隐藏（0）。要想移动或调整窗口的大小，你可以使用窗口参考调试工具。



```
Do(  
  DisplayDebugger(1),  
  Maximize(Debugger)  
)
```

描述

如果你打开调试工具，你将看到以下的窗口。



这个界面被分为三个部分，从上之下为：工具栏，源代码显示框和信息。我们将对每个进行描述。

第一部分：工具栏

工具栏在调试工具窗口的顶端，它有七个按钮和一个信息区，如下所示：

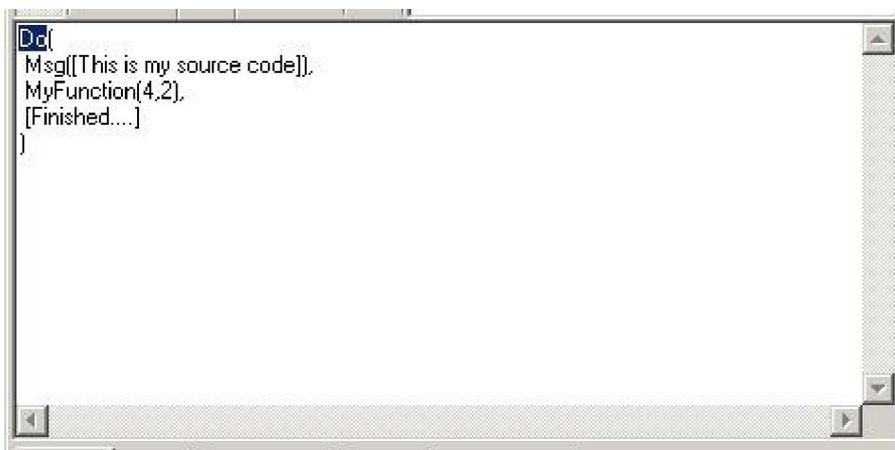


从左至右，按钮的功能如下：

-  切换此按钮以打开和关闭调试工具。如图所示调试工具处于打开状态（下压）。如果调试工具被关闭，那么其他按钮均为灰色（中止）；
-  运行。按下此按钮继续正常的执行。按{F5}有同样的效果；
-  停止。当有代码等待执行时，按下此按钮执行将中止。当系统处于闲暇状态时，按下此按钮下一个代码的执行将会被中断；
-  取消。按下此按钮将中止现行文本的执行。如果调试工具没有正在执行任何代码，那么这个按钮没有任何作用。使用这个按钮时要小心，因为现行的执行被完全中止了，但是如果以不受限制的速度模拟时，下一个事件将立即被执行。
-  步进。当调试一连串指令时，这个按钮强制调试工具进入指令中去，并且调试内部指令。通常这些其他的指令都是一个指令的参数。按{F8}有同样的效果。
-  跳过。当调试一连串指令时，这个按钮强制调试工具跳过现行指令内部运用的指令。按{Shift+F8}有同样的效果。
-  更新。按下此按钮更新窗口底端的信息页面中所示的信息。有哪些页面，这些页面包含什么信息和它如何运行这些问题在窗口的第三部分（‘信息页面’）中有所描述。在一些页面上可以使用键盘按下此按钮，比如在环境页面中，如下所示。

第二部分：源代码显示框

这个部分在调试工具窗口的中间，如下所示：



它显示你正在对4DScript代码进行调试。这个代码与编辑器中的代码不同，因为它产生于调试工具。这就解释了为什么外观会自动生成，评语被忽略，代码与所进行的执行指令中的形式相同。这意味着属性名称可以编译为索引属性，数学运算可以重组，原子标签和‘:=’指令可以编译为功能。虽然看上去不同，但结果是一样的。

如果选中的部分是一个单一的词，如图所示，那么这个词就是即将被执行的指令。如果选中的部分是一连串指令，那么这一串指令会被执行，但是调试工具会让你看到结果。结果显示在结果页面中，窗口的第三部分（‘信息页面’）将有描述。

选择单一的词和选择一连串指令之间的差别可见如下的例子：

EXAMPLE

我们将显示代码，当调试：

$2*(3+4)$

进入两次以后，它会显示如下：



调试工具正准备执行 ‘+’ 这个加法功能，作为 ‘*’ 这个乘法功能的参数。现在我们进行三次进入，或是跳过一次，这时两个参数 ‘3’ 和 ‘4’ 都被执行。现在所选择的内容如图所示：



这意味着 ‘+’ 功能被执行，并且结果将会返回至 ‘*’ 功能。我们进一步进入，我们将会看到 ‘*’ 也会被完成：



用户不能改变显示框以内的代码，但是你可以改变选中的部分，就像在其他任何编辑器中一样。然而，改变了选择对象并没有任何效果。

第三部分：信息页面：

信息页面在调试工具窗口的底端。每个页面都有与之相连的标签，并有相应的名称，选择标签即可显示相应的页面。标签如图所示：



所有的页面都包含信息，并且有时最好重新调整页面的高度。你可以把鼠标放置到源代码显示框和标签之间，当鼠标指针变成一种调整型指针时，按住鼠标左键并向上移动鼠标，直到页面足够高时便可以松开鼠标左键。



为了说明为什么和怎样使用每个页面，我们按照从左向右的顺序对每个页面进行描述。

- 选项：



选项页面有两组选项：调试和视窗。

后一组视窗选项中包含有显示和隐藏相应页面的开关。打勾的选项表示该页面可用并且相应的标签有显示。未打勾的选项表示该页面不可用并且相应的标签被隐藏。在此图中，所有的选项都被选中，所以所有的页面都可用。选项标签不能被隐藏。

前一组调试选项中包含有控制你想要调试的代码种类的开关。这些代码种类有：

1. 仿真 代码

这是默认的调试选项。这种包括原子事件 onEvent, onEntering, onEntered, onExiting, onCreate, onDestruction, onReset, onOcready, onlcready, onMessage 和 onInit。如果此种选项处于打开状态，那么所有的仿真都能被调试。

2. 动画 代码

这种包括2d绘图和4d绘图的原子事件。如果此选项被打开，那么动画就能被调试，但是至少要有一个动画窗口。动画窗口经常被刷新，所以不要打开这一选择，除非你真的想要调试动画。

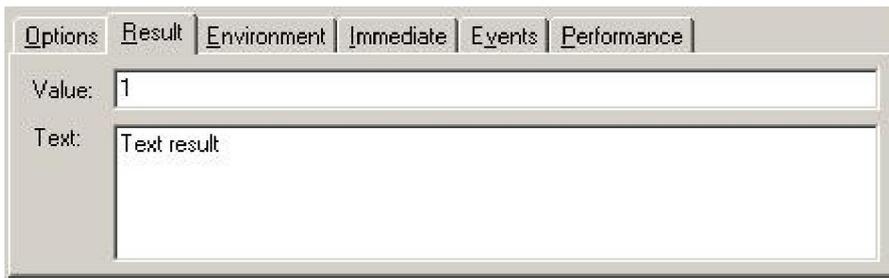
3. 用户事件 代码

这种包括所有必须被执行的代码，它们产生于用户行为，比如选中菜单中的一个选项，按下加速按钮，还有原子用户使用事件，互动窗口中所输入的代码和所有的图形用户界面事件。

4. 不可调试 代码

这种包括所有的内部代码，如动态数据交换 (DDE-) 和对象链接嵌入式(OLE-)-回调，和文件只读代码。回调不能被调试，因为它会中断其他应用程序。同样如下所述的动态页面中的代码也不能被调试。

• 结果：

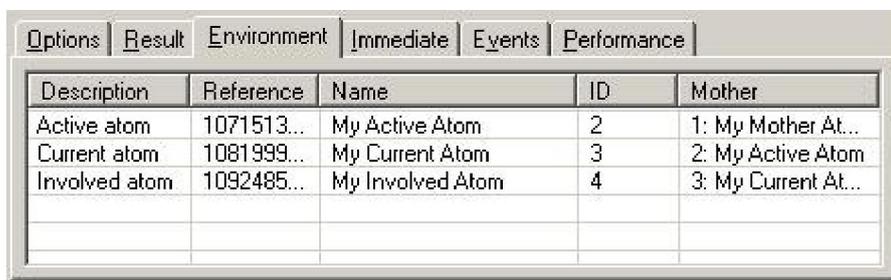


这个页面包含了上一个已执行指令的结果。结果的数值显示在上面的空格里，结果的文本显示在下面的空格里。在此图中两个结果都有显示：数值为1，文本为‘文本结果’。

每次完成一个指令时结果都会被更新。在源代码显示框中这一指令将不会再被选择：这种选择指的是下一个待命的指令，或者是使用这一指令的指令。

结果是不可能改变的，因为结果已经返回到呼叫指令了。你可以使用这个页面监控结果，同时进入4DScript代码，比如，当你想要找出‘奇怪行为’的原因时。

• 环境：



Description	Reference	Name	ID	Mother
Active atom	1071513...	My Active Atom	2	1: My Mother At...
Current atom	1081999...	My Current Atom	3	2: My Active Atom
Involved atom	1092485...	My Involved Atom	4	3: My Current At...

这一页面包含了原子信息。其中只有活动原子(4DScript 词条 a)，现行原子(c)，和涉及原子(i)可用，并且顺序是固定的。

如果原子存在，则显示：

- 原子的意义参考，与执行相应的4Dscript 词条的结果相同，
- 原子的名称，
- 原子的ID，和
- 母原子的ID和名称，如果存在的话。

原子参考有可能是无效的，比如当所涉及的原子已经在事件代码中某处被损坏时。在那种情况下就会显示‘无效参考’。

每次进入或跳过之后信息都会被更新，但是也可以通过点击工具栏中的更新按钮进行手动更新。

这种手动更新同样可以按{F5}完成，但仅当环境信息处于启动状态时，这时显示如下：

Description	Reference	Name	ID	Mother
Active atom	1071513...	My Active Atom	2	1: My Mother At...
Current atom	1081999...	My Current Atom	3	2: My Active Atom
Involved atom	1092485...	My Involved Atom	4	3: My Current At...

所有栏的大小都可以通过选择标头的边缘来调整，使该栏达到想要的宽度。双击这些边缘中的一个就能自动调整此栏的大小，能容纳所有信息。

Description	Reference	Name	ID	Mother
Active atom	107151359	My Active Atom	2	1: My Mother At...
Current atom	108199935	My Current Atom	3	2: My Active Atom
Involved atom	109248511	aMy Involved Atom	4	3: My Current At...

如果没能显示全部的栏，底部就会出现滚动条。当调整其他栏或整个窗口时滚动条又会消失。

- 动态：



这一页面看上去与互动窗口相同。它的运行方式也相同，甚至{F5}键也可用来执行进入代码，但是仅当代码编辑器处于启动状态时。

一些不同点也存在。

- 当调试一个指令时，互动窗口也同样被执行，但是环境页面中显示的原子参考 不能被改变。如果在动态页面中执行了相同的代码，那么这些参考也会改变。

- 动态页面中所输入的代码永远都不能被调试。如果代码是无限循环的，那么它不能被中断。

如果你在调试某个代码的同时还想改变原子参考，那么这一页面会十分有用。

- 事件：

Time	Code	Priority	Atom ID	Name	Involved
1	1	0	2	My Active Atom	No Invol...
1.5	2	0	3	My Current Atom	4
3	2	1000	5	My New Atom 2	No Invol...
3	1	0	2	My Active Atom	3
9	3	0	4	My New Atom 1	3

这一页面显示了有关等待事件的信息。这些信息包括时间，事件代码，优先事件，原子的ID和名称以及所涉及原子的ID。如果没有涉及其中的原子，那么会显示‘无涉及原子’。

这一列表不能自动更新，所以必须进行手动更新。你可以按工具栏中的更新按钮或是按{F5}键，但仅当事件列表处于开启状态时，此时显示如下：

Time	Code	Priority	Atom ID	Name	Involved
1	1	0	2	My Active Atom	No Invol...
1.5	2	0	3	My Current Atom	4

此列表可以通过按标头‘时间’，‘代码’，‘优先’，‘原子ID’，‘名称’或‘涉及’来分类。通过这种方式搜索原子的事件便可以简化。

所有栏都可以通过选择标头的边框来调整大小，如环境页面中所述一样。

- 执行：



此页面有助于监控仿真的执行。它有两个重要的部分：用于仿真vs.动画的时间分析（处理器时间），和4Dscript用法分析（计数命令）。

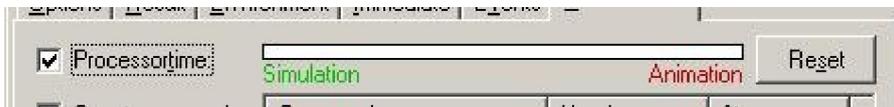
如果你想自动更新此页面，可选择更新频率：关闭（不更新），很低（每十秒一次），低（每五秒一次），正常（每两秒一次），高（每秒一次）或极高（每秒两次）。在默认情况下，更新频率为关闭。

通过按{F5}可以进行手动更新，但仅当页面处于开启状态时。

如上所述，这一页面有两个部分，使得我们能以三种不同的方式监控执行情况：

1. 执行监控：仿真vs. 动画

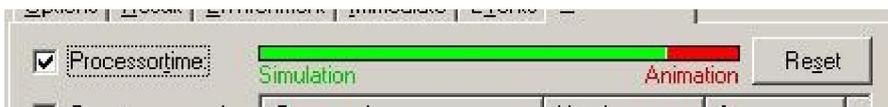
在开始分析之前，确保处理器时间选项被选中，此时页面显示如下：



使用它最好的方式是选择无限仿真速度，并且仅打开一个动画窗口和仅运行仿真。

EXAMPLE

首先我们打开执行页面上的处理器时间选项，并开始仿真。一段时间之后你的分析便显示如下：



结论：大部分时间都是用于仿真，只有一小部分是用于动画。

然而，执行页面是使用计数器来监控执行过程的。所以如果你运行的时间很长，那么仿真计数器变化的频率会低于动画计数器。因此，如果你的模型在进行了数小时的仿真之后速度减慢，那么它将不会出现在此页面中。你必须按重启键重新开始计数：此时绿色和红色栏又会变成白色。如果此页面已被更新，那么它应该显示如上述例子。

EXAMPLE

我们将进行与上一个例子相同的操作。但是如果出现了以下显示：

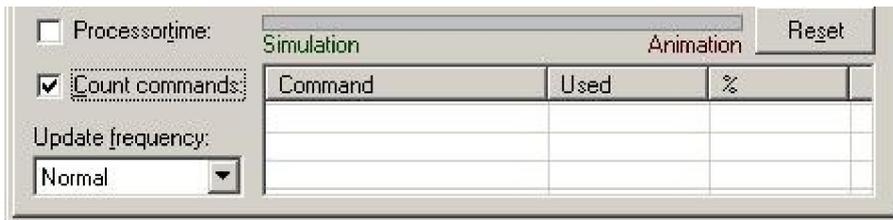


结论：大部分时间都用于动画，而不是仿真。如果你没有选择无限仿真速度，那么这种情况经常会发生。但是如果你是以无限仿真速度在运行的话，这意味着动画消耗时间过多，所以此时最好关闭其中的某些部分，或是完全关闭。

下面继续说明此页面中的第二部分，命令计数：

2. 执行监控：4D Script用法

我们从简单的例子开始。如果你使用了如上所述的处理器时间选项，那么首先请将它关掉。为了打开每个4Dscript词条的计数器，只需要保证计数命令选项被选中即可。此时页面应该显示如下：



无论何时只要页面被更新，使用过的词条计数都会被放入至列表中。如果词条没有出现在列表中，它就被添加至底端。命令栏包含了词条，使用栏中显示了此词条被使用的次数。最后一栏为空。数值，文本和变量也同样被放入至此列表中。过一会此列表中便会出现数据。此时列表显示如下：



要想显示更多的词条，而不是仅有如图中所示的两个词条，只需调整信息页面的大小即可，与信息页面开头所描述的方法相同。

你可以通过点击标头‘命令’或者‘使用’来将数据分类。命令是按字母顺序分类的，计数器是按数字从高到低的顺序排列的（最高的在第一位）。如果数据以‘...’结尾，这时你可以通过选择标头的边框重新调整栏的大小，这样栏中就会足够宽的空间显示出此数据，这与环境页面中所描述的方法相同。

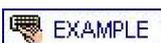
计数器将持续增长，所以一段时间之后你应该使用重启按钮刷新列表，使得计数器再次从零开始计数。

列表可以给你提供一些有关4D Script 词条用法的信息。你会发现，很多词条都只在动画打开时被使用，而其他词条只在仿真运行时被使用。但是如果我們想知道哪些命令消耗了大量时间时又怎样呢：

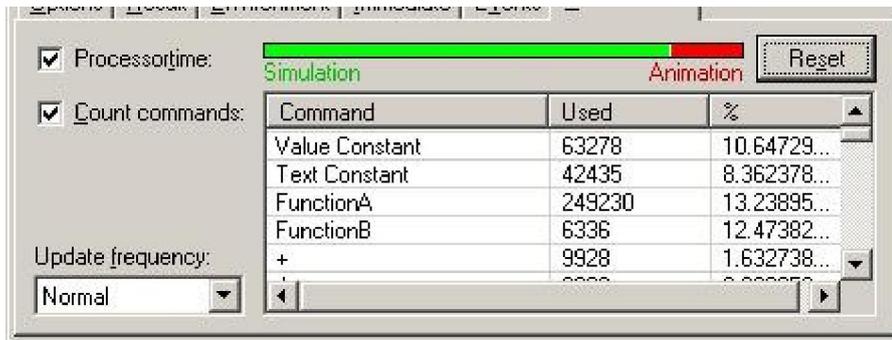
3. 执行监控：4D Script 处理器时间用法

结合如上所述的两项功能，我们可以跟踪用于每个指令的时间。同时打开处理器时间选项和计数命令即可。有些计数器有可能在两个选项被打开之前就已经增加了。如果出现这种情况，调试工具便不会跟踪每个指令的处理时间。因此按下重启按钮确保所有的计数器都显示为零。

过了一会列表便会有显示，并且%栏也会出现数据。对每个词条而言，这一栏显示的是其所用时间占进行所有执行所需总时间的比例。



首先我们同时打开处理器时间选项和计数器命令。我们按下重启按钮确保所有的计数器显示为零。一段时间之后，执行页面将显示如下：



‘功能A’的执行次数为249230，占总时间的13.238953...%。我们将其与‘功能B’作比较，‘功能B’的执行次数仅为6336，但占总时间的12.47382...%。结论：每执行一次功能A比执行一次功能B要快。

现在三栏都有数据显示，并且这三栏都能通过标头‘命令’，‘使用’或者‘%’来分类。命令是按字母顺序分类，而计数器和比例是按数字从高到低排列。

当持续使用这些计数器而没有重启时，就会出现极小的数值。比如0.0000012345678%：这个数字将会显示为1.2345678E-6。此时‘E’会被隐藏起来，因为栏宽太小。这样按比例排序时看上去会很奇怪，因为这样的数值看上去比0.1这样的数值要大，但是0.1却会出现在1.2345...之上的某个地方。

为了使得所有的数字都可见，只需通过选择标头的边框来调整栏框的大小即可，如环境页面中所述一样。

调试工具窗口的所有部分都已经有所描述。但是还有更多的特色：

当调试其他4D Script 的同时执行4DScript

你可以在调试一些代码的同时执行其他的4DScript代码。比如，当你正在调试一些仿真代码时，菜单中的所有项目仍在进行。这意味着你必须小心：



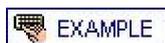
确保你所执行的代码不会影响到你正在或即将调试的代码。这会引发不可预测的结果，诸如违规访问和无效指针运算。所以当你调试仿真或动画时，不要重启新模型，不要关闭任何动画窗口之类的。

防止这种情况最好的办法便是使用工具栏里的开关选项关闭调试工具。如果你想中止仿真过程，只需在动态页面中输入Stop（停止）并按Execute（执行）按钮。然后等待至它完成执行。指令Exit（退出）可以用来立即终止执行并关闭ED，不会出现任何警告。关闭调试工具的另一种方法是点击工具栏中的Cancel（取消）按钮。这样将中止整串指令的执行。

如果你的动画过程耗费了太多时间，那么调试时不要关闭任何动画窗口，而应该按照如上所述步骤进行操作，直至ED不再调试时再最后关闭动画窗口。接着等待所有窗口都关闭。

我怎样创建一个断点？

你可以使用4Dscript 进入调试工具来设置断点。不管这个命令何时被执行，都应检查调试工具是否处于打开状态。如果你使用一个不等于零的参数的命令，那么此时调试工具不得不开。即使调试工具不可见，它也可以显示出来，并且你可以对4DScript 代码开始调试，紧接进入调试工具命令之后被执行。



```
Do(
  Trace([This code is executed, without interruption]),
  EnterDebugger(1),
  Trace([This is the first statement to debug])
)
```

任意停止执行

如果调试工具处于打开状态，同时按下{Ctrl+Break}会强迫调试工具立即停止执行。然后你可以使用调试工具穿过现行的4DScript 代码，或者终止指令的执行。后面这个在你进入无限的循环时很有帮助。

受保护的代码

如果你打开一个加密文件，里面的代码是受保护的。这意味着你不能对它进行调试。然而，你可以按下{Ctrl+Break}来终止它。调试工具会提示你是否终止此执行或是继续。穿过此代码是不可能的。

这也同样适用于读取一个文件时所执行的代码的情况。你可以终止代码，但是你不能对它进行调试。如果在受保护的代码中存在一个突破点，强迫调试工具启动（如例子），那么错误监控器便会出现，并会显示错误‘不能进入调试工具’。