



操作指南



操作指南 2.0

DroidPilot 用户指南

本手册及附带的软件和其他文档受中国和国际版权法保护，并且只能依据附带的许可协议使用。软件功能及 DroidPilot 徽标是 深圳云迈科技有限公司或其子公司在中国和/ 或其他国家（地区）的商标或注册商标。如果本列表中缺少某商标，不能视为深圳云迈科技对此商标的知识产权弃权。

所有其他的公司、品牌和产品名均为其各自所有者的注册商标或商标。深圳云迈科技有限公司对于商标所属权问题拒绝承担任何责任。深圳云迈科技随时都有更新文档内容的权利而不必事先通知所有用户，用户使用请以网站下载版本为最新。本文档中所涉及的图有可能与实际工具不同，请以用户手中工具版本为准，深圳云迈科技随时有更新本文档截图的权利。

DroidPilot 深圳研发中心

地址：广东省深圳市南山区高新技术产业园科技中二路中钢大厦 M6 栋

@ 2012 深圳云迈科技，保留所有权利

如果您对此文档有任何意见或建议，请通过电子邮件发送至 Support@DroidPilot.com。

目录

欢迎使用 DroidPilot 客户端.....	6
使用本指南.....	6
文档更新.....	6
第 I 部分 测试流程.....	7
第 II 部分 使用测试对象.....	11
了解测试对象和对象属性.....	11
了解测试对象.....	11
使用对象属性浏览器查看对象属性.....	11
在脚本编辑器中使用对象方法和方法语法.....	11
第 III 部分 创建测试组件.....	13
设计测试和组件.....	13
关于设计测试和组件.....	13
计划测试或组件.....	13
制作测试或组件.....	14
创建项目.....	14
创建组件.....	14
捕获对象.....	15
设计测试步骤.....	16
生成脚本.....	16
编译脚本.....	17
增强测试或组件.....	17
了解检查点.....	18
关于了解检查点.....	18
将检查点添加到关键字视图或脚本.....	18
了解检查点类型.....	19
检查对象属性值.....	19
关于检查对象属性值.....	19
创建标准检查点.....	20
了解“检查点属性”设置对话框.....	22
修改检查点.....	23
检查表和数据库.....	24
检查文本.....	24
参数化值.....	25
关于参数化值.....	25
参数化步骤和检查点中的值.....	25
使用组件配置参数.....	27
使用项目配置参数.....	28
使用数据表参数.....	29
输出值.....	30
配置值（版本更新中）.....	31
配置常量和参数值.....	31
添加包含编程逻辑的步骤.....	32

使用条件语句.....	32
添加注释.....	33
第 IV 部分 运行调试测试组件	35
调试脚本.....	35
语法检查.....	35
使用“从步骤运行”命令	35
停止运行会话.....	35
修改脚本.....	36
运行脚本.....	36
单个脚本调试运行	37
运行批处理测试 (*)	37
使用测试报告.....	39
查看测试结果.....	39
关于查看测试报告	39
了解“测试结果”窗口	39
查看运行会话的结果.....	40
查看检查点结果.....	41
查看参数化值和输出值结果.....	42
删除测试结果.....	43
导出测试报告.....	43
第 V 部分 配置 DroidPilot 客户端	45
单一设备管理.....	45
多设备管理.....	46
第 VI 部分 使用支持 DroidPilot 环境	47
使用应用部署工具.....	47
部署在多台移动设备.....	49
第 VII 部分 附录	50
附录 I - DroidPilot 安装指南	50
附录 II - 了解基本的 VBScript 语法	56

欢迎使用 DroidPilot 客户端

欢迎使用 Android 应用自动化关键字驱动测试解决方案 DroidPilot。本文档提供快速创建并运行测试和组件所需的所有内容。

使用本指南

本指南描述如何使用 DroidPilot 测试应用程序。它提供的逐步说明可以帮助用户创建、调试和运行测试及组件，并在测试过程中报告检测到的缺陷。

它包括下列部分：

第 I 部分 启动测试流程

概述 DroidPilot 和测试流程的主要阶段。

第 II 部分 使用测试对象

介绍 DroidPilot 如何识别移动应用程序中的对象以及如何使用对象库。

第 III 部分 创建测试组件

描述如何创建测试项目及组件、插入检查点和参数以及输出值、使用正则表达式、使用操作以及处理运行会话期间发生的异常事件。

第 IV 部分 运行及调试测试和组件

描述如何运行测试脚本及组件、分析结果以及控制运行会话或组件脚本中的错误。

第 V 部分 配置 DroidPilot 客户端

描述如何修改 DroidPilot 设置以满足测试需要。

第 VI 部分 使用支持 DroidPilot 的环境

描述使用 DroidPilot 的特定环境信息

第 VII 部分 附录

有关问题的参考信息

DroidPilot 文档集

除本用户指南以外，DroidPilot 还会附带下列附录文档：

《DroidPilot **安装指南**》介绍如何安装 DroidPilot

文档更新

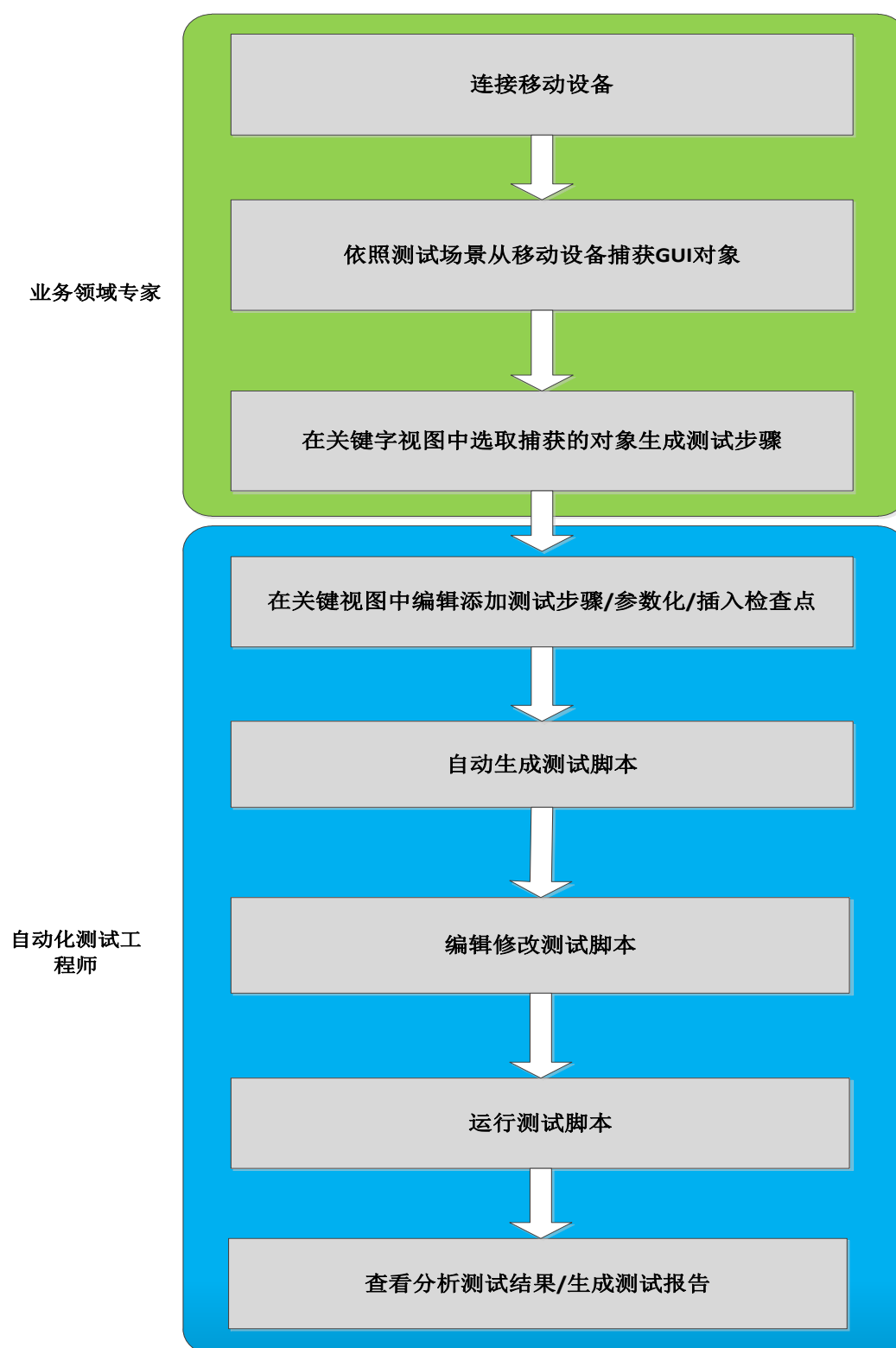
DroidPilot 会继续使用新信息来更新其产品文档。可以从客户支持网站

(http://www.droidpilot.com/support_ch/documents_ch.aspx) 上下载该文档的最新版本。

要下载更新文档，请执行下列操作：

- 1 在产品网站中，单击“Documentation”链接。
2. 选择“DroidPilot Operation Manual”。

第 I 部分 测试流程



简介

欢迎使用 DroidPilot Mobile Application 高级关键字驱动测试解决方案。通过 DroidPilot 可以测试标准的 移动应用对象、Native 移动应用程序。您可以根据 Unicode 标准 (<http://www.unicode.org/standard/standard.html>) 的需求兼容 Unicode，使您可以在三种语言环境下测试应用程序。Unicode 使用 8 位或 16 位代码值表示所需字符，从而可以处理和显示多种语言和字符集。本简介提供下列 DroidPilot 功能和测试过程的概述：

- 使用 DroidPilot designer 进行测试
- 了解测试流程
- 在脚本编辑器视图中编程
- 使用 TestingPilot 管理测试流程
- 修改许可证信息

使用 DroidPilot Designer 进行测试

DroidPilot 通过关键字驱动的测试方法模拟在应用程序中执行的操作，可以方便地创建测试和业务组件。测试和业务组件是两种不同类型的文档，可用于测试您的应用程序是否按预期工作：

测试：组织成一个或多个操作的步骤集合，用于验证应用程序是否按预期执行。

业务组件：表示应用程序中单任务的步骤集合。业务组件（也称为组件）在多设备运行中心，由业务流程测试组合为特定的场景以建立业务流程测试移动应用程序时，DroidPilot Designer 将创建您执行的每个步骤，并生成在基于表的关键字视图中图形化显示这些步骤的测试或组件。例如，单击按钮、选中复选框、或输入文本编辑框值。

按照业务测试步骤设计完成后，您可以利用检查点检查应用程序中特定对象的属性。例如，可以指示 DroidPilot 检查特定文本字符串是否在特定位置显示，还可以检查数据库中的数据与应用显示的值是否相同。

可以通过添加或修改关键字视图中的步骤来进一步增强您的测试或组件。当您点击“生成脚本”快速的生成测试脚本后，可以运行测试脚本，DroidPilot 将执行测试或组件中的每个步骤。

运行会话结束之后，您可以查看详细列出所执行的步骤以及哪些成功哪些失败的测试结果。

测试由操作组成，添加到测试中的步骤都包含于测试的操作中。

了解测试流程

使用 DroidPilot 测试包括三个主要阶段：

创建测试或组件

您可以通过对移动应用程序捕获 GUI 对象并使用关键字驱动功能向关键字视图中手动添加测试步骤来创建测试或组件。然后，可以在专业脚本编辑器中使用特殊的编程语句来修改您的测试或组件生成的脚本。

创建测试或组件：

- 以下方式向测试中添加步骤：
- 捕获建立对象库并使用这些对象在关键字视图或脚本视图中手动添加步骤。

建立一个项目获取在应用程序所有对象的对象库。有关建立对象库的详细信息，请参阅捕获测试对象”可以通过在关键字视图中选择项目和操作并输入所需信息创建步骤。有关详细信息，请参阅“设计测试步骤”。高级用户可在脚本编辑器视图中添加步骤。有关详细信息，请参阅“[调试脚本](#)”。

- 在测试或组件中插入检查点。

检查点检查页面、对象或文本字符串中的特定值或特征，通过它可以标识应用程序是否正常运行。有关详细信息，请参阅“[了解检查点](#)”。

启动测试流程

- 通过用参数替换固定值拓宽测试或组件的范围。

测试移动应用程序时，可以参数化测试或组件以检查应用程序如何使用不同数据执行相同的操作。您可能提供数据表中的数据，定义环境变量和值，定义测试、组件或操作参数和值，参数化测试或组件时，DroidPilot 将用参数代替测试或组件中的固定值。

当使用数据表参数时，DroidPilot 将为测试或者操作的每次循环使用数据表中不同行的值。每一个使用不同参数化数据集的运行会话称为一次循环。有关详细信息，请参阅“[参数化值](#)”也可以使用输出值从测试或组件中提取数据。输出值是在运行会话过程中检索并输入数据表格或另存为变量或参数的值。您可以随后使用该输出值作为测试或组件的输入数据。这样就可以在该测试或组件的其他部分使用运行会话中检索到的数据。有关详细信息，请参阅第 13 章“输出值”。

- 使用 DroidPilot 中众多的功能测试功能来增强您的测试或组件，或添加编写语句来实现更复杂的测试目标。

运行测试或组件

创建测试或组件后，请运行它。

- 运行测试或组件检查您的移动应用程序。

测试或组件将从其第一行开始运行直至测试或组件结束时停止。在运行中，DroidPilot 将连接到您的应用程序，执行测试或组件中的每一项操作，检查所有指定的文本字符串、对象或数据库表。如果使用数据表参数对测试进行了参数化，DroidPilot 将对您定义的每组数据值重复该测试（或测试中的特定操作）。有关详细信息，请参阅“[运行测试和组件](#)”。

- 运行测试或者组件以对其进行调试。

可以控制运行会话，帮助您标识和消除测试或组件中的缺陷。使用“脚本行执行”命令可运行调试部分及单个语句的测试或组件。还可以设置断点使测试或组件运行到在预定点停止。每当测试或组件在断点处停止时，可以在“调试查看器”中查看其变量的值。有关详细信息，请参阅“[调试脚本](#)”。分析结果:运行测试或组件之后，您可以查看其结果。

- 在“结果”窗口中查看结果。

运行测试或组件之后，您可以在“测试结果”窗口中查看运行的结果。您既可以查看结果的概要也可以查看每个详细报告。有关详细信息，请参阅“[查看测试结果](#)”。

- 报告在运行会话过程中检测到的缺陷。

如果已经安装了测试管理平台 TestingPilot，可以将发现的缺陷报告给数据库。您可以指示 DroidPilot 自动报告测试或组件中每个失败的步骤，也可以从“测试结果”窗口手动报告。TestingPilot 是一个集中的测试管理解决方案。有关详细信息，请参阅“TestingPilot”用户手册。

在脚本编辑器视图中编程

可以使用脚本编辑器视图选项卡查看测试或组件的基于文本的版本。测试或组件由使用类似 VBScript 语法编写的语句组成，这些语句对应于关键字视图中显示的步骤和检查。有关测试或组件中可用的测试对象和方法，以及如何使用 VBScript 编程的详细信息，请参阅《测试脚本参考》（选择“帮助”>“DroidPilot 帮助”）。

使用 TestingPilot 管理测试流程

您可以将 DroidPilot 和 TestingPilot 一起使用，后者为集中测试管理解决方案。可以使用 TestingPilot 管理移动自动测试项目以及报告并跟踪缺陷。还可以创建报告和图，帮助您在软件发布前查看测试计划、测试脚本的运行和缺陷跟踪的进度。使用 QuickTest 时，可以创建测试或组件并直接将其保存到 Quality Center 项目。有关详细信息，请参阅“TestingPilot 用户手册”。

修改许可证信息

安装 DroidPilot 之后，您再安装许可证号。在使用期限到期后或者可以改变您的许可证类型。可以联系 DroidPilot 客户支持网站上的信息购买新的许可证。请求许可证的网站的 URL 是

<http://www.droidpilot.com/en/support>

如果要购买许可证。有关详细信息，请参阅产品网站。要安装部署，请参阅《[DroidPilot 安装指南](#)》

第 II 部分 使用测试对象

了解测试对象和对象属性

本章描述 DroidPilot 如何记住和标识应用程序中的对象，解释测试对象和运行时对象的概念，并解释如何查看对象的可用方法和相应的语法，从而便于您在关键字视图中向脚本添加语句。

本章描述：

- 了解测试对象
- 使用对象属性浏览器查看对象属性
- 在脚本编辑器中使用对象方法和方法语法

了解测试对象

DroidPilot 的测试对象是被测移动应用的界面组件，元素，以及一些窗口小部件（**widget**），通过部署在 PC 上的 DroidPilot Client 可以轻松捕获移动设备上每一屏（Android Activity）上的这些元素，组件，部件，所有组件，元素，窗口部件属性值都直接来自移动设备上应用的界面数据，您可以轻松在关键视图中依照测试场景进行使用。

使用对象属性浏览器查看对象属性

使用“对象浏览器”，可以查看打开的应用程序中任何对象的属性。可以单击对象树上的任何对象，使用“属性”显示其属性和属性值。

要查看对象属性，请执行下列操作：

1. 打开应用程序，抓取对象页面。
2. 展开项目树，进入含有要查看的对象的界面。
3. 选择一个对象，单击鼠标右键 > “属性”。或者打开属性视图（默认 workbench 是打开的属性视图），还可以从“关键字视图”中鼠标右键单击“Insert”，在插入标准检查点对话框中，也列出了选中对象的所有属性选项。

提示：可以使用 **Object** 属性检索“对象浏览器”中显示的运行时属性的值。

在脚本编辑器中使用对象方法和方法语法

在语法编辑器中我们可以使用 **GetROProperty** 方法：获取当前对象的某一个属性值，一般用于条件语句内：例如：

```
If Window("MainActivity-3").TextView("list_content_title").GetROProperty("Text")="共[count]张"
Then
    Reporter.ReportEvent micPass,"The count is correct","This message is user defined"
Else
    Reporter.ReportEvent micFail,"The count is incorrect","This message is user defined"
```

End If

第 III 部分 创建测试组件

设计测试和组件

通过对各个项目的业务模块划分测试组件，抓取你在 Android 上应用程序上运行此业务所执行的操作界面，然后选取对象元素到关键字视图窗就可以创建测试脚本了。创建测试或组件之后，可以使用检查点和其他特殊测试选项来增强它。

本章描述：

- 关于设计测试和组件
- 计划测试或组件
- 制作测试或组件
- 增强测试或组件
- 管理测试

关于设计测试和组件

在 DroidPilot，项目，组件，操作页面及对象元素是通过项目树的形式层级展现，通过对选中的组件，首先抓取要测试的操作页面，其次选取对象元素来设计测试步骤，然后生成自动测试脚本。抓取对象时 DroidPilot 以图片方式展现，您可以通过选择图片上的每个对象，添加到测试步骤设计窗，将要执行的每个“步骤”显示为关键字视图中的行。步骤是您依照测试用例所做的、能够更改应用程序中的页面或对象内容的任何事情，例如单击链接或在编辑框中键入数据。

在关键字视图中，您可以将检查点插入到测试步骤设计中。“检查点”会将您对所需测试对象中捕获的元素的设置值，与在运行会话期间捕获的同一元素的值进行比较。这有助于确定您的应用程序是否正常运行。

当您测试应用程序时，您可能希望检查它是如何用不同的数据执行相同操作的。这称为“参数化”测试或组件。您可以通过定义环境变量和值在“数据表”中提供数据（如 CSV 文件）。有关详细信息，请参阅“[参数化值](#)”。

在获取对象后，可以通过在关键字视图中添加和修改步骤，或者直接修改脚本来进一步增强您的测试或组件。

计划测试或组件

在开始抓取对象之前，应该计划您的测试或组件，并考虑下列建议和选项：

- 确定要测试的业务类型。根据不同的业务建立不同的项目，如针对 Android SmartPhone，Tablet 检查操作系统中应用的功能。
- 划分要测试的业务模块。根据不同的业务模块建立不同的组件，如针对 Android 系统要检查名片簿，记事本，计算器，行业使用的等应用程序。
- 确定要测试的功能。根据要测试的功能来抓取对应的操作页面。
- 确定要在运行测试或组件过程中检查的信息。检查点可以检查应用程序中的文本字符串、对象和表的差异。有关详细信息，请参阅“[了解检查点](#)”。

- 评估需要制作的事件类型。与执行多个任务的长测试或组件相比，检查应用程序的特定功能或者完成一个事务的短测试或组件要更好。
- 考虑通过用参数替换固定值来增加测试或组件的能力和灵活性。当您参数化测试或组件时，可以检查它是如何用多组数据（或由外部源存储或生成的数据）执行相同操作的。有关详细信息，请参阅“参数化值”。
- 链接到 Runner 测试并从 TestingPilot 测试或组件中调用测试计划或者使用 DroidPilot AutoRunner([运行批测试脚本](#))

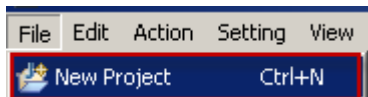
制作测试或组件

在开始抓取对象之前，应该按您计划的测试或组件先建立好项目和组件。DroidPilot 根据选中的组件抓取需执行的每个操作页面，然后将对象添加到关键字视图中将它们显示为步骤，并在脚本中生成它们。


注意，默认情况下，每项测试包括一项操作，但每项测试可以包括多项操作。

组件不使用操作。每个组件都是一个独立的单位，并且不能包含对其他组件的调用。

创建项目



要创建项目，请执行下列操作：


1. 打开 DroidPilot。有关详细信息，请参阅“启动 DroidPilot”。
2. 新建项目：
 - 要新建项目，请单击“新建项目”按钮，或者选择“文件” > “新建项目”。
 - 要打开现有的项目，现有的项目默认是随系统打开显示在左边项目树上，请单击此项目，右键菜单 > “打开项目”按钮，或者选择“文件” > “打开项目”。
 - 要导入一个新项目，选择“文件” > “导入项目”。在“导入项目”对话框中，选择一个导入项目类型，然后单击“下一步”，指定导入项目的路径，选择项目文件，单击“完成”。
3. 输入项目名字
4. 指定项目路径
5. 单击“确定”，新建的项目名字显示在左边项目树上，完成项目创建。

创建组件

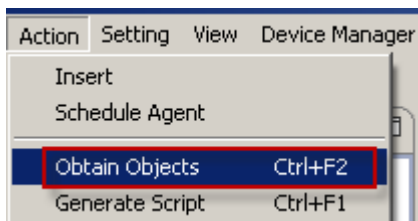


要创建组件，请执行下列操作：

1. 打开 DroidPilot。有关详细信息，请参阅“启动 DroidPilot”。

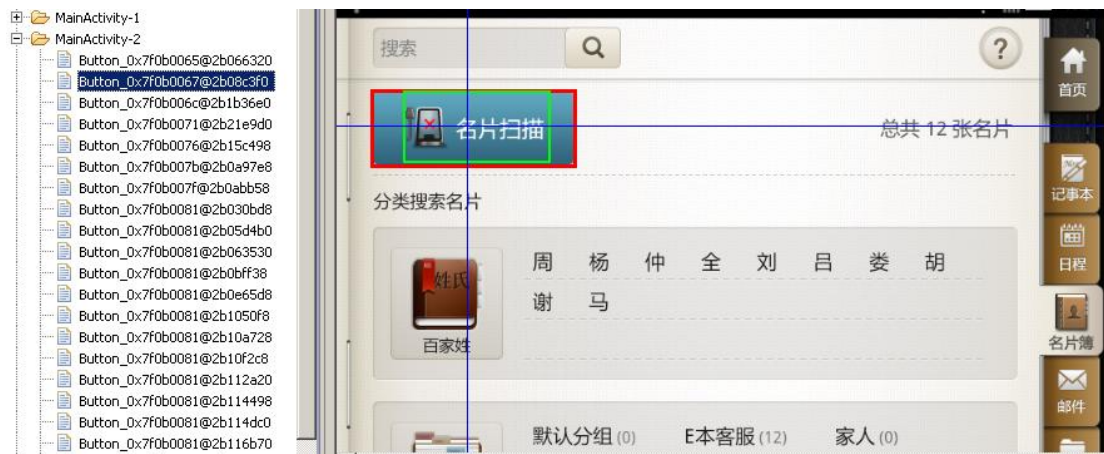
2. 新建组件: 
- 要新建组件, 请单击“新建组件”按钮, 或者选择“文件” > “新建组件”。
- 或者选中一个项目, 右键菜单 > “新建组件”
3. 选择为哪个项目新建组件, 已经选择的项目默认被选中。
4. 输入组件名字
5. 单击“确定”, 新建立的组件名字显示在左边项目树上对应的项目节点下面, 完成组件创建。

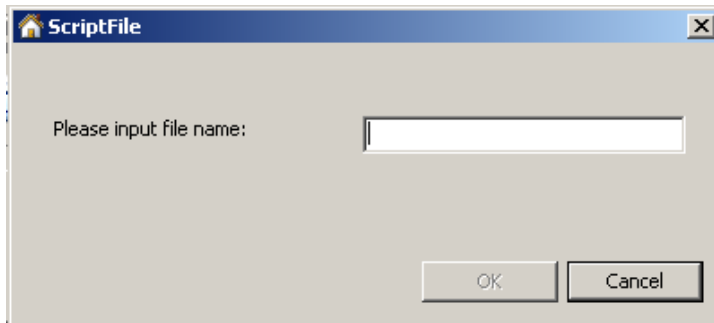
捕获对象



要创建组件, 请执行下列操作:

1. 打开 DroidPilot。有关详细信息, 请参阅“启动 DroidPilot”。
2. 选中一个项目下面的组件。
3. 单击“抓取对象”按钮, 或者选择“活动” > “抓取对象”。
4. 弹出窗口里选择连接的设备。
5. 窗口“获取对象进行中。。。”, 当进度到 100%时, 提示窗消失。
6. 抓取的操作页面显示在选择的组件节点下面, 默认命名为“MainActivity[N]”, 后面数字自动递增。您可以自己重命名。
7. 选中此页面节点, 图片自动显示在右边浏览窗口。单击图片上的对象元素, 可以自动对应到左边项目树上对象名称被高亮显示。

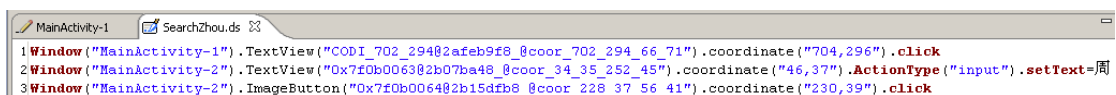




- 脚本生成成功后，显示在此组件下面的脚本“Script”节点下。

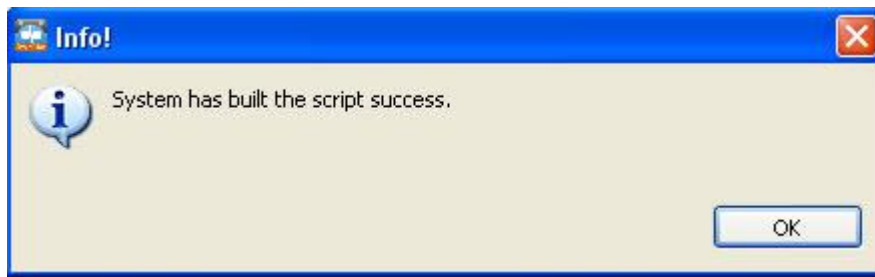


- 脚本文件被打开显示在右边的脚本编辑器，并列于关键字视图窗口。



编译脚本

选中所生成的脚本，点击“活动” > “编译脚本”，编译完成后有提示信息显示编译完成。当运行脚本时默认情况下会自动编译脚本。



增强测试或组件

在创建测试步骤或脚本以后，可以使用多种选项来增强它。

可以向测试步骤中添加检查点。“检查点”是测试事件中的一个步骤，它将在运行会话过程中指定的属性值与为该测试事件内的相同测试对象属性存储的值进行比较。这使您可以确定应用程序是否正常运行。有关创建检查点的详细信息，请参阅 [“了解检查点”](#)。

您可以参数化测试事件，以便在运行会话过程中将固定值替换为来自外部源的值。值可以来自“数据表”、您所定义的环境变量或者 DroidPilot 在运行会话过程中生成的值。有关详细信息，请参阅 [“参数化值”](#)。

可以从测试或组件中检索值并将其作为输出值存储在“数据表”中。然后，您可以将这些值作为测试或组件中的输入参数。这使您可以将在运行测试或组件过程中检索的数据用于该测试或组件的其他部分。有关详细信息，请参阅 [“输出值”](#)。

您可以使用特殊的 DroidPilot 选项，通过编程语句来增强测试事件。步骤生成器可逐步引导您完成测试事件的整个测试过程。您还可以对测试进行休眠同步，确保应用程序已准备好以便 DroidPilot 执行测试中的下一个步骤，并且您可以通过定义和度量事务来度量应用程序执行测试中的步骤所需要的时间量。有关详细信息，请参阅“[添加包含编程逻辑的步骤](#)”。

您还可以在脚本编辑器中手动输入 VBScript 语句以及使用 DroidPilot 测试对象和方法的语句。

了解检查点

关于了解检查点

通过 DroidPilot 可以将检查点添加到设计测试步骤的关键视图或脚本中。“检查点”是将指定属性的当前值与该属性的期望值进行比较的验证点。这使您能够确定应用程序是否正常运行。当添加检查点时，DroidPilot 会根据用户的选择将检查点添加到步骤设计窗中的当前行之前或之后，并在步骤设计窗中添加一条“检查检查点”语句。当您运行测试或组件时，DroidPilot 会将检查点的期望结果与当前结果进行比较。如果结果不匹配，检查点就会失败。可以在“测试结果”窗口中查看检查点的测试结果。

注意：添加的检查点会在脚本里自动生成如下脚本，如果您希望检索检查点的返回值（一个布尔值，指示该检查点是通过还是失败），您可以在检查点设置框输入期望的值，必须是小写，如果在脚本里添加，必须保证等号“=”二边不能有空格。例如：

```
Window("MultiGroup").Button("android.widget.Button@2b17e270").checkPoint=<(isEnabled,true)>
```

有关脚本语法的详细信息，请参阅附录“了解基本的 VBScript 语法”。

将检查点添加到关键字视图或脚本

您可以在设计测试步骤过程中或在编辑测试脚本时添加检查点。在基本测试步骤的关键字设计好之后，通常可以更为方便地定义检查。

有几种方法可以添加检查点。

在设计测试步骤时添加检查点

- 在关键视图中选中一个对象所在的步骤行，使用右键菜单上的“插入”命令，或者单击菜单“活动” > “插入”选项去插入标准检查点
- 弹出的检查点属性列表将显示与设计测试步骤窗的选定步骤相关的所有检查点选项。

只在编辑脚本时添加检查点

- 在编辑脚本时，利用关键字弹出下拉列表选项添加检查点。
- 在脚本里当输入关键字后或在关键字后输入点“.”号会弹出下拉列表选项供选择，这是非常方便或加快用户手工写脚本的方式。

注意：

如果使用编辑脚本的方法，请确保插入的有关待检查对象的属性名称正确，若不清楚属性名称的可以到“插入”菜单弹出的检查点列表查看。

了解检查点类型

您可以插入下列检查点类型以检查应用程序中的各种对象。

➤ “标准检查点”检查应用程序中对象的属性值。标准检查点会检查各种对象，如按钮、单选按钮、组合框、列表等等。例如，您可以检查在选择单选按钮之后它是否处于激活状态，或者可以检查编辑字段的值。

标准检查点在所有加载项环境中都受到支持（请参阅第 N 页的“受支持的检查点”）。

有关标准检查点的详细信息，请参阅“检查对象属性值”。

➤ “存在检查点”检查应用程序中的对象是否存在。例如，您可以检查所选对象被删除后是否还存在于页面上。

➤ “数据库检查点”检查应用程序中的统计数据是否正确。例如，您可以检查页面上所显示记录条数是否跟数据库里的一致。

➤ “文本检查点”检查文本字符串的内容是否正确。例如，您可以检查页面是否显示的文本是否您预期的内容

➤ “逻辑检查点”插入一些带业务逻辑的判断语句检查是否按照这个业务逻辑执行。例如，当有数据时或加入新的数据后这个按钮才可用，然后点此按钮进行下一步操作等。

检查对象属性值

通过向测试或组件中添加标准检查点，可以对不同版本的应用程序或网站中的对象属性值进行比较。

本章描述：

- 关于检查对象属性值
- 创建标准检查点
- 了解“检查点属性”设置对话框
- 修改检查点

关于检查对象属性值

可以使用标准检查点来检查应用程序中的对象属性值。标准检查点将对抓取操作页面期间捕获的对象属性的预期值，与运行会话期间对象的当前值进行比较；可以使用标准检查点对应用程序的图像、表、按钮属性以及其它对象执行检查。

注意：用户可以为所有受支持的测试环境创建标准检查点

创建标准检查点

通过向测试或组件中添加标准检查点，可对应用程序中具有预期属性值的指定对象进行检查。要设置标准检查点的选项，可使用“检查点属性”对话框。

创建标准检查点

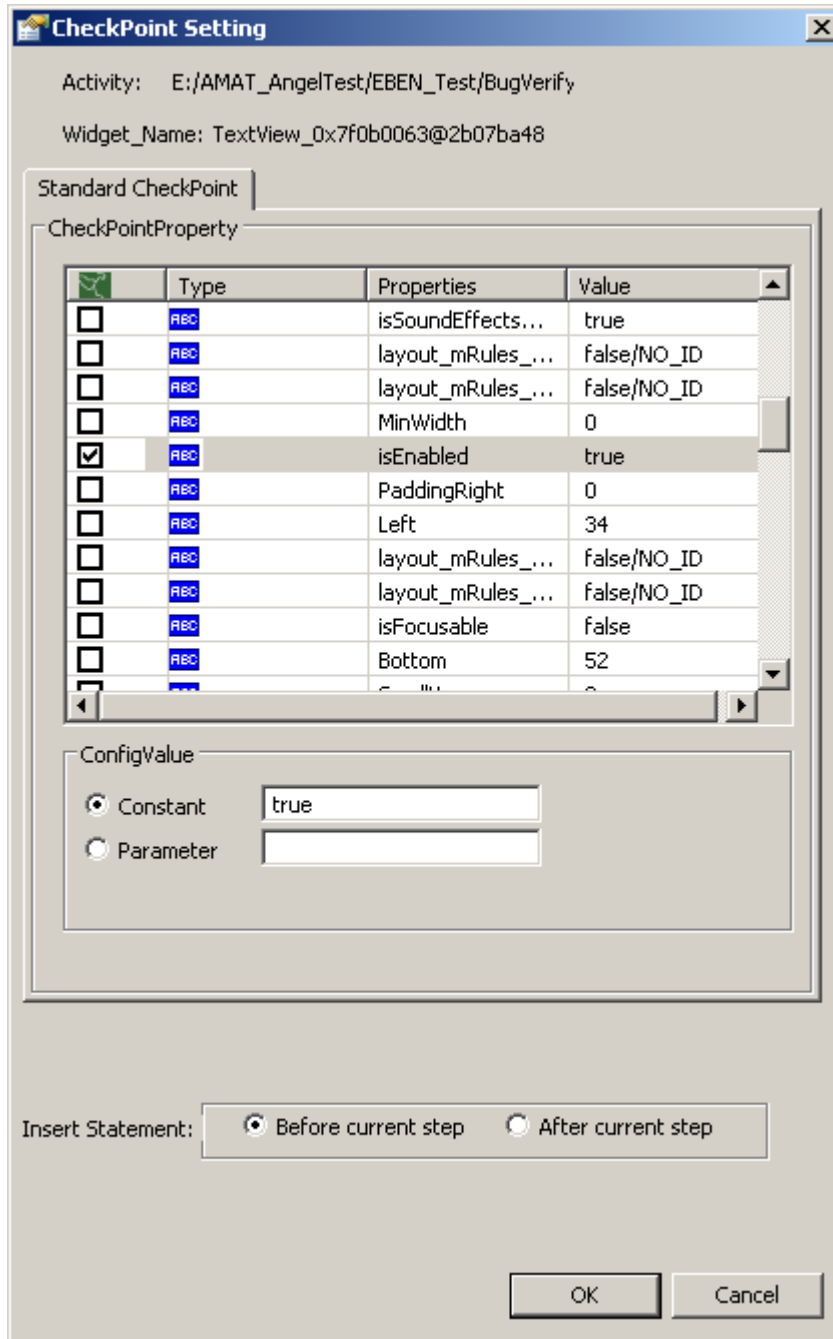
用户可以在设计测试步骤关键字视图中或编辑测试脚本时添加标准检查点。

要在设计测试步骤关键字视图中添加标准检查点，请执行下列操作：

1. 在关键视图中选中一个对象所在的步骤行，鼠标右键单击“插入”选项，或选择菜单“活动” > “插入”。

注意：如果用户要检查的对象是在一个浮动或弹出窗口时，在抓图时此窗口也必须抓出来。

2. 单击要检查的属性。属性的默认值（抓取对象时获得的值）被显示到 ConfigValue 的 Constant 输入框，用户可以在此修改希望此属性值在运行会话过程中期望的值。有关详细信息，请参阅第 N 页的“了解“检查点属性”设置对话框”。
3. 或者用户也可以选择期望的值要来自于设置的参数。
4. 设置好属性值后，选择 Insert Statement 选项。
5. 单击“确定”关闭对话框。在关键字视图中，将为选定对象添加一行检查点步骤，当编译脚本后将添加一行检查点语句。



要在编辑脚本时添加标准检查点，请执行下列操作：

1. 在测试脚本中找到对象所在的执行步骤行，确定是在当前步骤之前检查对象属性的值，还是当前步骤之后检查对象属性的值。
2. 首先输入抓取此对象所在的操作页面的名字，例如输入“Window (” 后会弹出对象页面下拉列表选项供选择。

```
Window (
    ("MainActivity-1")
    ("MainActivity-2")
    ("MainActivity-3")
    ("MainActivity-4")
    ("MainActivity-5")
    ("MainActivity-6")
```

3. 在操作页面后输入“.” 后会弹出对象下拉列表选项供选择。

```
26 Window("AddColumn").
```

```
.Button("android.widget.Button@2b1ead98")
.CardLimitLayout("com.ebensz.widget.CardLimitLayout@2b17b240")
.CheckedTextView("android.widget.CheckedTextView@2b0670f0")
.CheckedTextView("android.widget.CheckedTextView@2b0de3c0")
.CheckedTextView("android.widget.CheckedTextView@2b140a18")
.CheckedTextView("android.widget.CheckedTextView@2b166690")
.CheckedTextView("android.widget.CheckedTextView@2b180b88")
.CheckedTextView("android.widget.CheckedTextView@2b181bc0")
.CheckedTextView("android.widget.CheckedTextView@2b185b10")
.CheckedTextView("android.widget.CheckedTextView@2b187bf8")
.CheckedTextView("android.widget.CheckedTextView@2b19f2a0")
.CheckedTextView("android.widget.CheckedTextView@2b2d9bc8")
.CheckedTextView("android.widget.CheckedTextView@2b467c00")
```

4. 在对象后面输入 “.” 后会弹出此对象对应的坐标及 checkPoint 选项供选择。

```
26 Window("AddColumn").Button("android.widget.Button@2b1ead98").
```

```
.Exist
.GetRoproperty("Text")
.checkPoint=<(name,value)>
.checked
.click
.selected
.longClick
.updateDate
.updateTime
.scrollTo
.viewExportedProperty
```

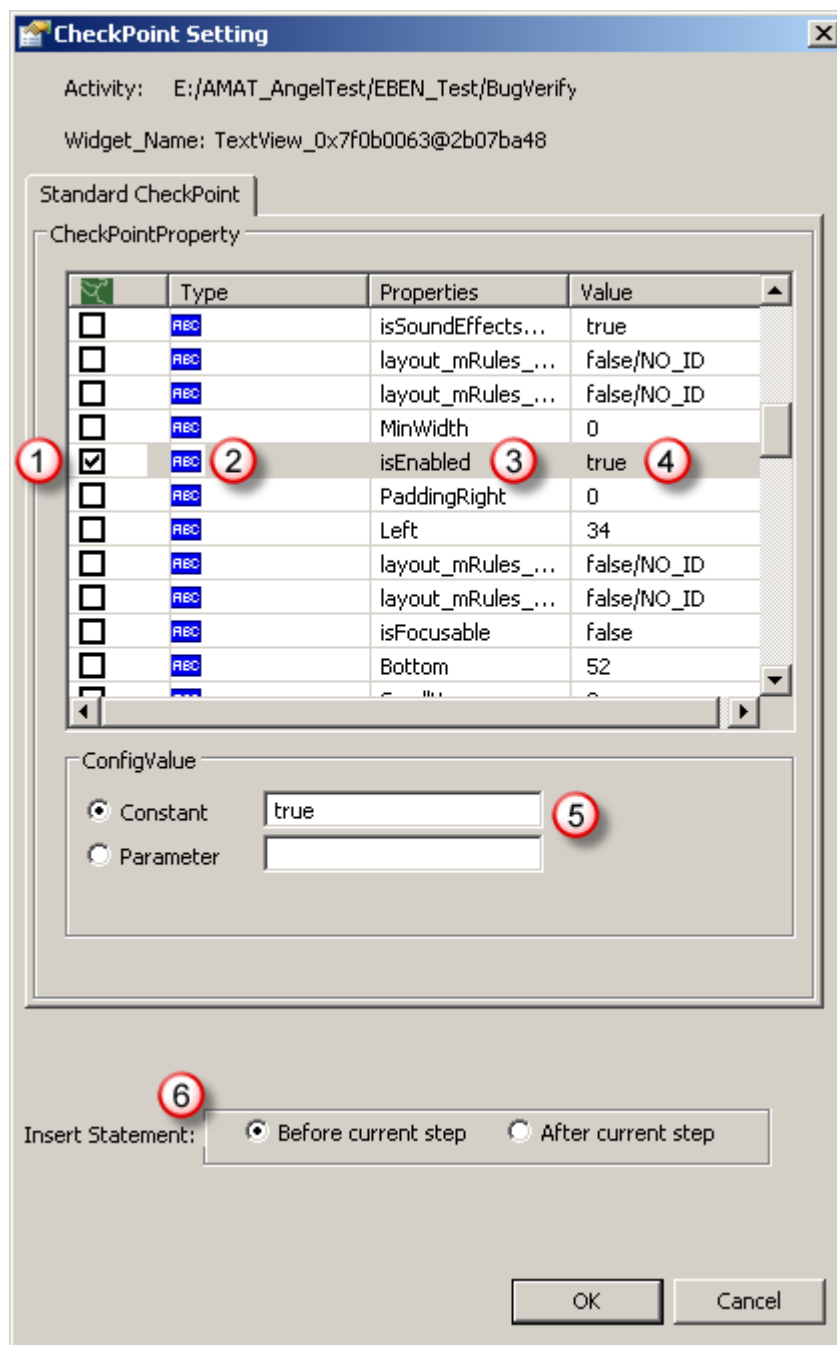
5. 选择 checkPoint 关键字，按要求的格式填入属性名字及期望的属性值。

```
Button("android.widget.Button@2b1ead98").checkPoint=<(name,value)>|
```

了解“检查点属性”设置对话框

在“检查点属性”设置对话框中，可以指定要检查对象的哪些属性并编辑这些属性的值。虽然根据所检查对象的类型，特定的元素会略有不同，但“检查点属性”对话框通常包含下列基本元素：

1. 选中的复选框表示将对该属性进行检查。
2. ABC 图标表示要检查的属性的值是一个常量。
3. 属性列显示当前选中对象所有相关的属性名称。
4. 值列对应到属性列对应的值，这是抓取对象时获得的值。
5. 配置当前属性在运行回话过程中期望得到的值。
6. 设置是在当前步骤之前或是之后执行此属性检查。



修改检查点

可以修改现有检查点的设置。例如，可以选择使用参数，也可以使用筛选器来指定要检查的图像源和链接。

要修改检查点，请执行下列操作：

1. 在关键字视图中单击带有检查点的一行，然后单击“值”列旁边的“#”按钮，在弹出的参数设置对话框修改值。
2. 修改属性值并单击“确定”。
3. 或者在关键字视图中单击带有检查点的一行，鼠标右键单击“删除”当前行。

4. 再在关键视图中选中一个对象所在的步骤行，鼠标右键单击“插入”选项，或选择菜单“活动” > “插入”。插入新的检查点。

检查表和数据库

可以在测试脚本中使用数据库检查点，检查通过应用程序访问的数据库以及检测缺陷。定义对数据库的查询，然后通过逻辑判断语句，变量比较和报表输出语句来检查查询的结果。

所有默认情况下受 DroidPilot 支持的环境以及多种外部加载项环境都支持数据库检查点。例如：

```
database (dbPath=mnt/sdcard/AppFiles/com.ebensz.cardname/card.db, pkgName=com.ebensz.cardname, dbName=card.db, sql=Select count(name) as countname From cards where spell like ? or head like ? , args=%A%/%A%, return=countname)
```

```
If "總共[countname]張名片"
"=Window("FamilyName2").TextView("android.widget.TextView@2b12b530").
GetROProperty("Text") Then
    Reporter.ReportEvent micPass, "SearchRecords", "Search by 'A'
correctly"
Else
    Reporter.ReportEvent micFail, "SearchRecords", "Search by 'A'
Incorrectly"
```

或者：

```
database (dbPath=mnt/sdcard/AppFiles/com.ebensz.cardname/card.db, pkgName=com.ebensz.cardname, dbName=card.db, sql=Select count(name) as countname From cards Where head like "%楊%" or spell like "%Y%", args=, return=countname)
End If
```

检查文本

通过将下列检查点之一添加到测试或组件，可以检查是否显示了指定的文本字符串。

- **文本检查点**– 通过它可以根据指定条件检查文本是否显示在屏幕、窗口或网页中。所有环境下都支持文本检查点。
- **标准检查点**– 通过它可以检查对象的 Text 属性。这是在许多 Windows Applications 中检查文本的首选方法。有关详细信息，请参阅第 16 页的“创建标准检查点”。
- **文本区域检查点**– 通过它可以根据指定的条件，检查文本字符串是否出现在 Windows Applications 的已定义区域。

参数化值

使用 DroidPilot 可以通过将固定值替换为参数，扩展基本测试或组件的范围。该过程（称为参数化）大大提高了测试或组件的功能和灵活性。

本章描述：

- 关于参数化值
- 参数化步骤和检查点中的值
- 使用组件配置参数
- 使用项目配置参数
- 使用数据表参数

关于参数化值

可在 DroidPilot 中使用参数功能，通过参数化测试或组件所使用的值来增强测试或组件。

参数是一种从外部数据源的变量。

可以参数化测试或组件中的步骤和检查点中的值。还可以参数化操作参数的值。

有二种方式配置参数：

- **配置组件参数**，通过它可以配置外部数据源指定的测试值，如 CSV 数据表指定的测试值。
- **配置项目参数**，通过它可以配置环境变量参数，随机数字参数或常量参数等。

有三种类型的参数：

- **常数变量参数**，通过它可以使用定义过的任何常量参数的值。
 - **数据表参数**，通过它可以创建使用来自外部数据源指定的测试值，指定您所提供的数据多次运行的数据驱动的测试（或操作）。在每次重复（或循环）中，DroidPilot 均使用数据表中不同的值。首先在组件参数中配置好数据源，然后在设计测试步骤关键字视图中设置参数。
- 例如，假设您的应用程序或网站包含一项功能，用户可以通过该功能从成员数据库中搜索联系信息。当用户输入某个成员的姓名时，将显示该成员的联系信息。可以参数化成员名称属性，以便在运行会话的每次循环期间，DroidPilot 可检查它如何用多组数据执行相同的操作。

参数化步骤和检查点中的值

设计或编辑测试步骤或脚本时，可以参数化步骤和检查点中的值。

可以参数化选定步骤的对象属性的值。还可以参数化为该步骤定义的操作（方法或函数参数）的值。

例如，您的应用程序可能包含一个带有编辑字段的表单，用户可以在该编辑字段中键入用户名。您可能希望测试应用程序是否读取该信息并将其正确显示在对话框中。可以插入一个对已登录的用户名使用内置环境变量的文本检查点，以检查显示的信息是否正确。

通过参数化检查点属性的值，可以检查应用程序如何基于不同的数据执行相同的操作。

例如，如果要测试按姓分组的名片簿，可以创建一个检查点，以便检查“王”姓组里名片数量是否正确。假设您需要检查针对各种不同姓里的名片数量是否正确。可以为姓组信息添加一个数据表参数，而不是为每个姓分别创建带有单独检查点的不同测试或组件。对于测试或组件的

每次循环，DroidPilot 都会针对不同的群组检查名片数量。

有关使用检查点的详细信息，请参阅“了解检查点”。

将某个值定义为参数时，需要首先选择该值，然后指定参数类型及其设置。

提示：使用步骤生成器添加新步骤时，可以参数化选定操作的值。

参数化对象和检查点的属性值

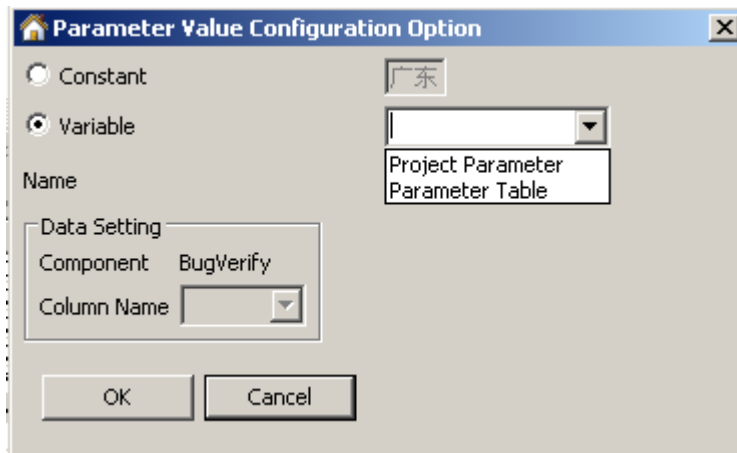
可以在“参数值配置选项”或插入“检查点属性”对话框中参数化对象的一个或多个属性的值。

可以在“脚本编辑器”中参数化检查点的一个或多个属性的值。

要参数化对象或检查点属性值，请执行下列操作：

1. 可以采用下列方式之一打开“参数值配置选项”对话框或“检查点属性”对话框：

➤ 选择“步骤” > “属性值”，单击属性值旁边的“#”按钮。将打开“参数值配置选项”对话框。



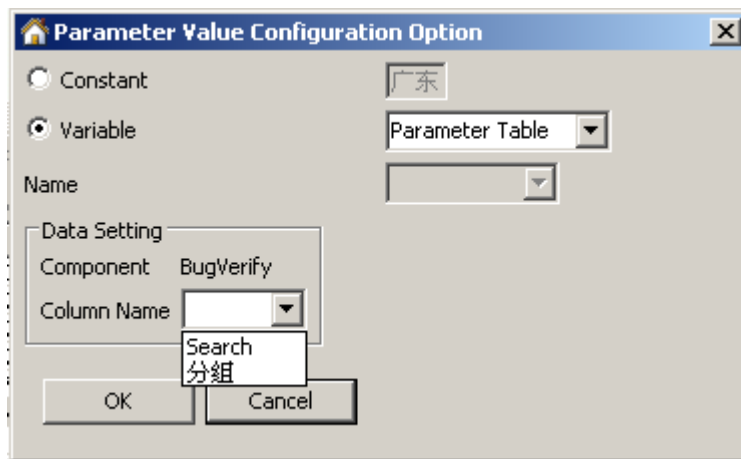
➤ 右键单击某个步骤并选择“InsertCheckPoint”。将打开“检查点属性”对话框。在此对话框可以设置参数值。

➤ 或选择“步骤” > 单击“活动” > “插入”菜单，将打开“检查点属性”对话框。

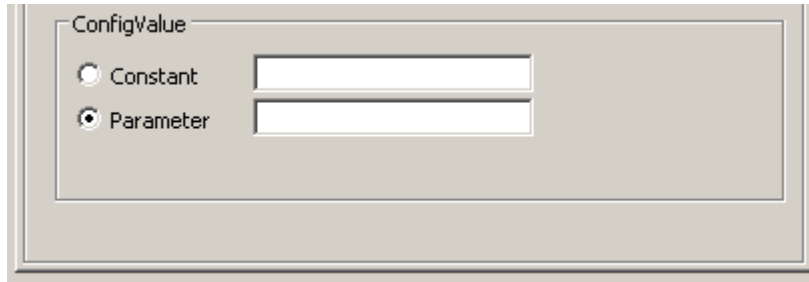
2. 选择要参数化的属性。

3. 在该对话框的“配置值”区域中，选中“参数”。

如果该值已经参数化，则“参数”框将显示该值的当前参数定义。如果该值尚未参数化，则“参数”框将显示该值的默认参数定义。



参数值配置选项：



检查点属性：

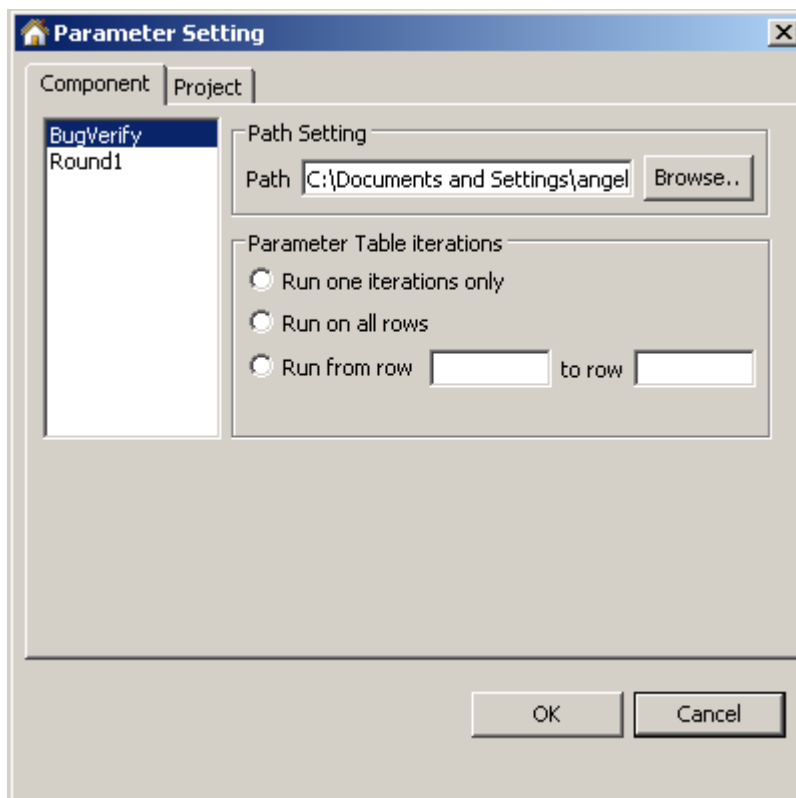
4. 接受或更改所显示的参数定义。

- 要接受显示的参数语句并关闭该对话框，请单击“确定”。
- 要接受显示的参数语句并参数化显示的其他值，请选择其他属性。
- 要更改参数类型或修改选定属性的值设置，请重新单击属性值旁边的“#”按钮。将打开所显示参数类型的“参数配置选项”对话框。

使用组件配置参数

要配置组件参数，请执行下列操作：

1. 选中一个项目，点击菜单“设置”>“参数设置”。
2. 选中一个组件。
3. 点击“浏览”设置外部数据源路径。
4. 设置重复运行次数。
 - 只运行一轮，表示只用数据表里第一行数据测试一次。
 - 运行所有行，表示用数据表里所有行数据测试。
 - 运行从第几行到第几行，表示用数据表里指定的第几行数据到第几行数据测试。
5. 点击“确定”保存数据。

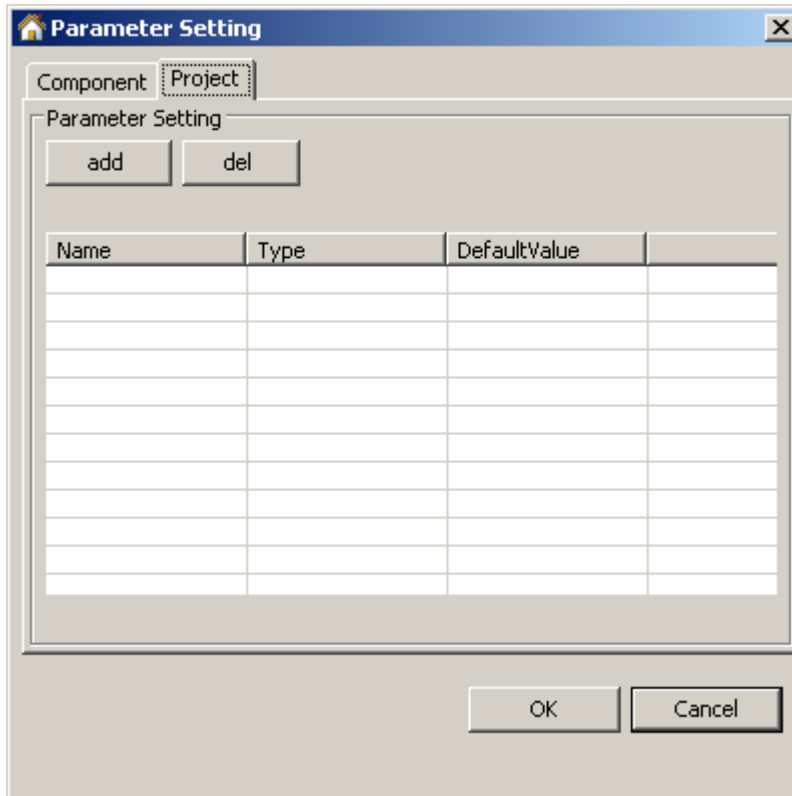


对这个组件的参数配置，将只用到这个组件本身。 各个组件之间配置的参数不能相互调用。

使用项目配置参数

要配置项目参数，请执行下列操作：

6. 选中一个项目，点击菜单“设置”>“参数设置”。
7. 选中“项目”标签页。
8. 点击“增加”按钮。
9. 设置增加项目参数的名字，参数类型，数据类型和默认值。
10. 参数类型有如下选择项：
 - 常量参数，表示使用设置的默认数据作为常量参数测试。
 - 环境变量参数，表示使用配置的本地或环境变量数据测试。具体信息请参见“使用环境变量参数”。
 - 随机数字参数，通过它可以插入随机数字作为测试或组件的值测试。具体信息请参见“使用随机数字参数”。
11. 点击“确定”保存数据。



对这个项目的参数配置，将用到整个项目。其下每个组件可以调用相同的项目参数。

使用数据表参数

可通过创建数据表参数来为参数提供可能的值列表。通过数据表参数可以创建使用您所提供的数据进行多次运行测试、组件或操作。在每次重复（或循环）中，DroidPilot 均使用数据表中不同的值。

例如，考虑重复性的查询或创建操作，不必创建大量查询或创建的提交来执行该测试。这是一个既费时又费力的低效解决方案。通过使用数据表参数，可以连续对多个查询或创建运行测试或组件。

参数化测试或组件时，需要首先抓取访问应用程序的页面并针对所请求的一条路线来设计步骤并检查查询或创建功能。

然后将创建的步骤里操作值替换为某个数据表参数，并在数据表中添加自己的数据集，每条路线对应一个列。例如，查询对应一列，创建对应一列。

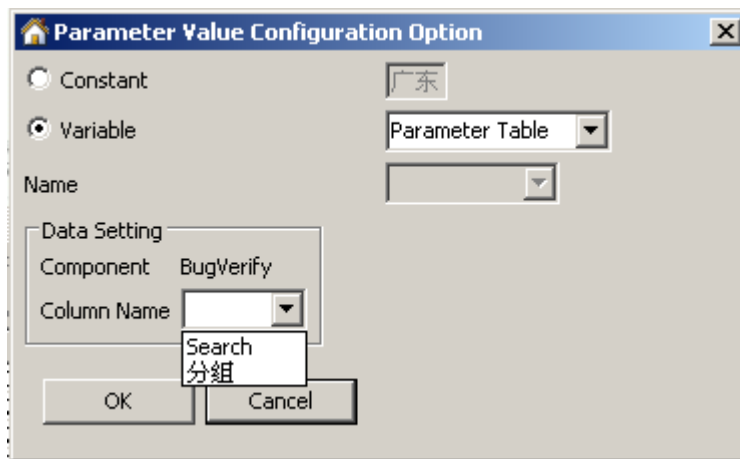
新建数据表参数时，将在数据表中添加新的一列，表中的每个列都表示单个数据表参数的值列表。列标题是参数名。

表中的每一行都表示 DroidPilot 在测试或组件的单个循环期间为所有参数提交的一组值。运行测试或组件时，DroidPilot 将针对表中的每一行数据运行一次测试或组件循环。例如，如果测试的数据表在整个列表中有十行，则运行十次循环。

	A	B
1	Search	分組
2	周	家人
3	正	朋友
4	王	同学
5	朝	同事
6	李	驴友

设置数据表参数选项

当选择“数据表”作为参数类型时，可以通过“参数配置选项”对话框将参数配置为使用数据表中的值。然后选择数据表中对应的列名。“值配置选项”对话框的“参数”部分与“参数选项”对话框非常相似。



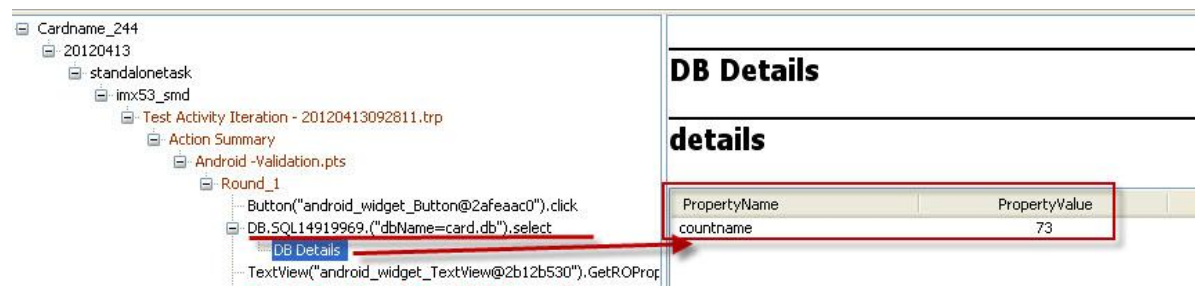
输出值

通过 DroidPilot 可以检索您的测试或组件中的检查值，这些值会自动输出到测试报告里存储。此后，您可以查阅这些值，并在运行会话的不同阶段使用该值作为输入。

标准检查点输出值示例：

数据库检查点输出值事例：

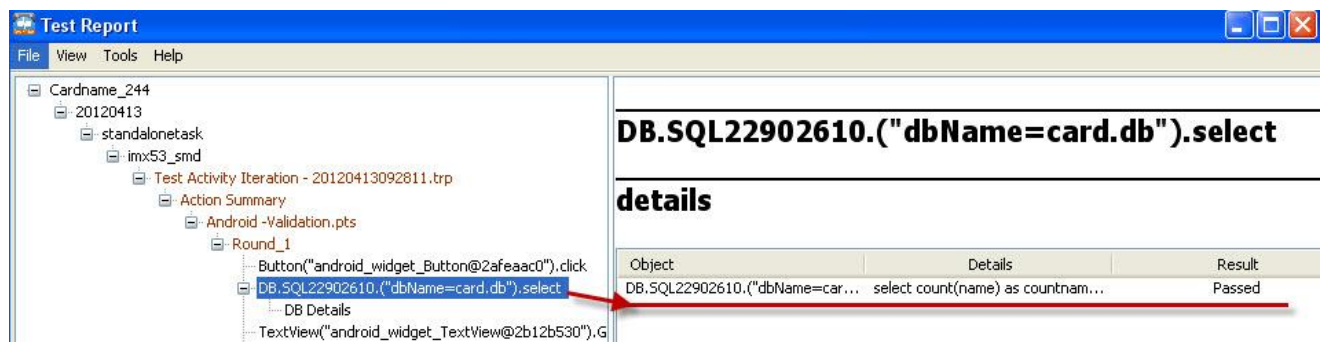
```
database (dbPath=mnt/sdcard/AppFiles/com.ebensz.cardname/card.db, pkgName=com.ebensz.cardname, dbName=card.db, sql=Select count (name) as countname From cards where spell like ? or head like ? , args=%A%/%A%, return=countname)
//從資料庫中查詢姓氏為 A 字母名片總數
```



自定义输出报告信息事例：

```

database (dbPath=mnt/sdcard/AppFiles/com.ebensz.cardname/card.db, pkgName=com.ebensz.cardname, dbName=card.db, sql=Select count(name) as countname From cards where spell like ? or head like ? ,args=%A%/%A%, return=countname)
If "總共[countname]張名片"
"=Window ("FamilyName2").TextView("android.widget.TextView@2b12b530").
GetROProperty ("Text") Then
    Reporter.ReportEvent micPass, "SearchRecords", "Search by 'A' correctly"
Else
    Reporter.ReportEvent micFail, "SearchRecords", "Search by 'A' Incorrectly"
End If
    
```



配置值（版本更新中）

DroidPilot 允许您通过将值定义为常量或参数来配置属性和其他项目的值。

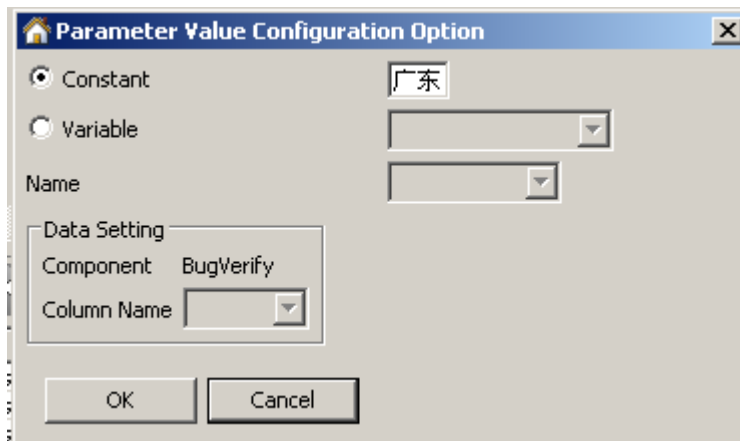
配置常量和参数值

通过以下方式可以将值定义为常量或参数：

- 在“参数设置”中设置组件参数，项目参数的来源，在“参数值配置选项”对话框中，单击选定值的参数化按钮，例如，在关键字视图或步骤生成器中。有关详细信息，请参阅第 22 页的“参数化对象和检查点的属性值”。
- 在对话框的“配置值”区域中，选择属性或参数，例如，在“检查点属性”对话框中。

在配置值区域中设置值（版本更新中）

在包含“配置值”区域的对话框中选择项目时，可以选择“常量”或“参数”来设置值。默认值为“常量”。



如果选择“常量”，则可以在“常量”框中直接编辑单行值。

配置参数值

如果对已参数化的值选择“参数”，则“参数”框显示该值当前的参数定义。如果对尚未参数化的值选择“参数”，则“参数”框显示该值默认的参数定义。

您可以单击“参数选项”按钮来选择不同的参数类型，或修改该值的参数设置。

对于显示的参数类型，将打开“参数选项”对话框。有关为特定参数类型定义值的详细信息，请参阅：

- “参数化步骤和检查点中的值”
- “使用组件配置参数”
- “使用项目配置参数”
- “使用数据表参数”

有关在测试和组件中使用参数的详细信息，请参阅“参数化值”。

添加包含编程逻辑的步骤

在设计测试步骤和生成脚本后，可以通过编辑脚本代码，使用编程语句来增强该测试或组件，即使用户在脚本编辑器视图中选择手动编程也是如此。

本章描述：

- 使用条件语句
- 将消息发送到测试结果
- 添加注释

使用条件语句

您可以使用条件语句来控制测试或组件的流程。使用条件语句 If...Then...Else，可以将决策过程纳入测试或组件中。

If...Then...Else 语句用于评估条件为真还是为假，根据结果，以指定运行一个或多个语句。条件通常是一个表达式，使用一个比较操作符将一个值或变量与另一个值或变量相比较。以下是可用的比较操作符： 小于<、小于或等于<=、大于>、大于或等于>=、不等于<>和等于=。

If...Then...Else 语句可以根据需要嵌入许多层。它具有以下语法：

If condition Then statements [Else elsestatements] End If

或者，可以使用块形式语法：

```
If condition Then
    [statements]
[ElseIf condition-n Then
    [elseifstatements] . . .
[Else
    [elsestatements]
End If
```

例如：

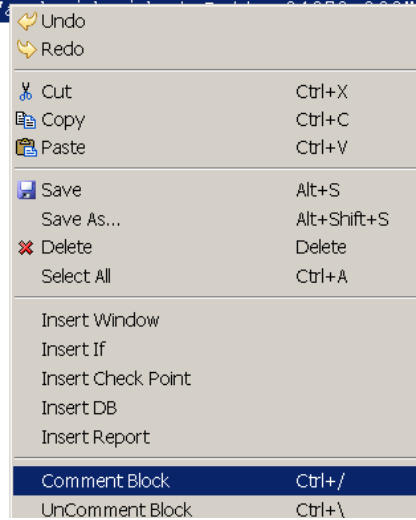
```
database (dbPath=mnt/sdcard/AppFiles/com.ebensz.cardname/card.db, pkgNa
me=com.ebensz.cardname, dbName=card.db, sql=Select count (name) as
countname From cards where spell like ? or head
like ? ,args=%A%/%A%, return=countname)
If "總共[countname]張名片
"=Window ("FamilyName2").TextView("android.widget.TextView@2b12b530").
GetROProperty ("Text") Then
    Reporter.ReportEvent micPass, "SearchRecords", "Search by 'A'
correctly"
Else
    Reporter.ReportEvent micFail, "SearchRecords", "Search by 'A'
Incorrectly"
End If
```

注意：有关条件语句的其他信息，请参阅 VBScript 文档

添加注释

您可以在脚本编辑器中使用脚本注释功能，脚本编辑器支持“//”，以及“/* */”的注释方式，用户也可以在右键菜单选择块注释。

```
Window("MainActivity-Carinfo1").EditText("android.widget.EditText@4089ef38").Action
//Window("MainActivity-Carinfo1").EditText("android.widget.EditText@4079e2e0").Acti
Window("MainActivity-Carinfo1").EditText("android.widget.EditText@4084ee20").Action
/* Window("MainActivity-Carinfo1").EditText("android.widget.EditText@407b6958").Act
Window("MainActivity-Carinfo1").Button("android.widget.Button@4089ef38").click*/
```



第 IV 部分 运行调试测试组件

调试脚本

通过对运行会话进行控制和调试，可以标识并消除测试和组件中的缺陷。

本章描述：

- 语法检查
- 使用“从步骤运行”命令
- 停止运行会话

语法检查

在创建测试或组件之后，应该检查该测试或组件的运行是否顺利，是否没有语法或逻辑错误。

要执行语法检查可以执行下列步骤：

1. 选中要执行语法检查的脚本。
2. 选择菜单“活动”->“语法检查”
3. 当有语法错误时会提示哪个语句出错，并显示行号。否则提示语法检查正确。

使用“从步骤运行”命令

DroidPilot 还可以使用“从步骤运行”功能对测试或组件进行调试。该功能从选定步骤开始运行测试或组件，直到结束。或者从脚本一开始运行测试，直到选择的步骤。

使用“从步骤。。。运行”功能可以执行下列步骤：

1. 打开脚本编辑器，
2. 点击鼠标右键，选择“运行从步骤。。。”，
3. 输入希望开始运行的脚本步骤行号，
4. 点击“OK”。脚本会从指定的行号开始运行直到结束。

使用“运行到步骤。。。 ”功能可以执行下列步骤：

1. 在脚本编辑器中，点击鼠标右键，
2. 选择“运行从步骤。。。”，
3. 输入希望开始运行的脚本步骤行号，点击“OK”。脚本会从指定的行号开始运行直到结束。

停止运行会话



选择“活动” > “停止”或单击工具栏上的“停止”按钮，可以停止运行会话。停止的测试或组件在前面执行完的步骤可以查看其执行结果。



要重新运行停止的测试或组件，可以单击“运行”按钮，也可以选择菜单“活动” >

“运行”，针对选择的脚本重新开始运行。

修改脚本

使用 DroidPilot 可以很方便的编辑脚本，它提供了强大的类似标准语言编辑器。当输入关键字后会自动弹出下拉菜单选项供选择。以帮助您快速的编辑和使用测试脚本。DroidPilot 支持的下拉选项如下：

1. 输入关键字“Window (” 后会弹出对象操作页面下拉列表选项供选择。

```
Window ({  
  ("MainActivity-1")  
  ("MainActivity-2")  
  ("MainActivity-3")  
  ("MainActivity-4")  
  ("MainActivity-5")  
  ("MainActivity-6")  
}
```

2. 在操作页面后输入“.”后会弹出对象元素下拉列表选项供选择。

```
Window {  
  ("A")  
  ("A-Detail")  
  ("A-Verify")  
  ("AddColumn")  
  ("DeleteGroup")  
  ("EditGroup")  
  ("EnterGroup")  
  ("FamilyName2")  
  ("FamilyNames")  
  ("HomePage")  
  ("MultiGroup")  
  ("NewGroup")  
  ("NewGroup")  
}
```

3. 在选中的对象后面输入“.”后会弹出此对象对应的所有操作选项供选择。

```
Button("android.widget.Button@2b1ead98").|
```

```
.Exist  
.GetProperty("Text")  
.checkPoint=<(name,value)>  
.checked  
.click  
.selected  
.longClick  
.updateDate  
.updateTime  
.scrollTo  
.viewExportedProperty
```

运行脚本

创建测试或组件后，可以运行它以检查应用程序的行为。DroidPilot 可以单个脚本调试运行，在 Devices Manager 中支持单个移动设备或进入“AutoRunner”批量脚本运行。

本章描述：

- 关于运行测试和组件
- 单个脚本调试运行

➤ 运行批处理测试

单个脚本调试运行

选中一个脚本点击“运行”按钮，或选择菜单“活动”->“运行”，弹出对话框里选择连接的设备，DroidPilot 将执行该脚本包含的所有步骤。如果已经定义了测试或组件参数，DroidPilot 将自动输入你设置好的值。当测试运行完成时，DroidPilot 将显示详细描述测试结果的报告。有关查看测试结果的详细信息，请参阅“查看测试结果”。

如果测试包含数据表参数，且设置为执行所有行，则 DroidPilot 为数据表中的每一行运行一次测试。每一个测试包含当前操作之前的所有步骤。

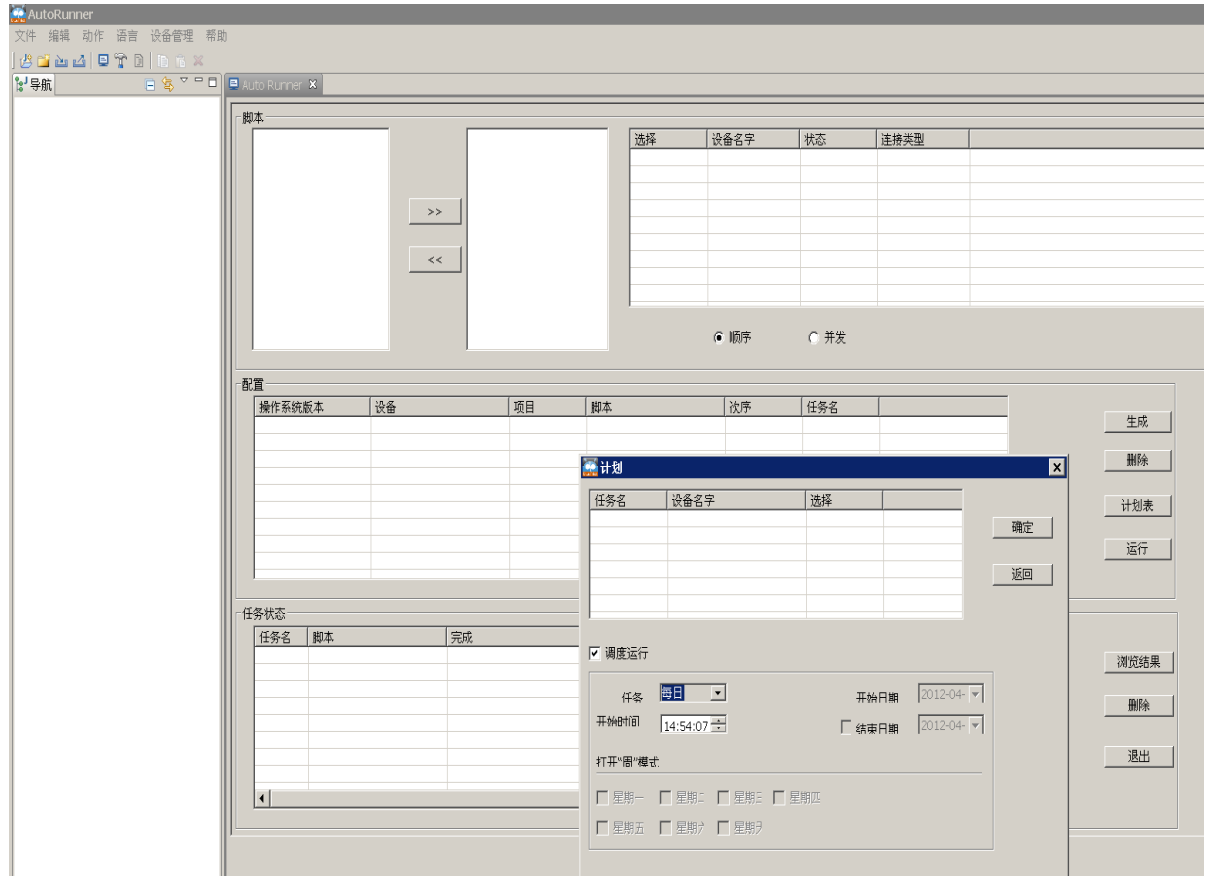
可以从头运行整个测试或组件，也可以运行测试或组件的一部分脚本。可以将某些脚本指定为“运行到当前行”，或“从当前行运行到某一行”脚本。以便让 DroidPilot 分批调试这些测试脚本。

运行批处理测试（*）

*该功能仅支援企业版用户

使用 DroidPilot Test AutoRunner，可以安装一批测试，计划执行时间，然后按顺序运行它们。使用 Test Tab，您可以设置测试列表，将该列表生成为配置文件，以便可以轻松地在另一个时间再次运行同一批测试。在批处理运行过程中，还可以选择在运行中包括或排除批处理列表中的测试。

DroidPilot AutoRunner 提供支持多设备的测试服务。可以一套测试运行于各种 Android 设备；或多套测试运行于多套设备；或多套测试脚本运行于一套设备。可以通过点击“停止”按钮随时停止批处理测试运行。



要设置和运行批处理测试，请执行下列操作：

1. 从“活动”菜单选择“Action（AutoRunner）”。将打开“Auto Runner”对话框。所有的项目，组件及脚本以树的形式展现在列表中供选择。首次使用需要使用者导入已经由 DroidPilot Designer 创建的专案及脚本。
2. 选择需测试的脚本，单击“添加”按钮。将添加到测试列表中。
3. 选择要在批处理测试列表中包括的测试设备。
4. 选择测试多台设备的方式。
 - 顺序，按顺序运行脚本在多台设备上。
 - 并发，并行运行单个脚本在多台设备上。
5. 单击“生成”按钮，将生成测试套配置文件，用户可以命名 Task Name。生成的测试套配置记录显示在配置区域的表格内。
6. 当准备好批处理测试套配置文件后，请单击“执行”按钮，计划批处理测试时间。
 - **Daily, Start Time, Start Date, End Date**，按指定的时间，在指定的起始和结束日期，每天运行此脚本。
 - **Weekly, Weeks On (Mon ~ Sun), Start Date, End Date**，按指定的周日，在指定的起始和结束日期，每周运行此脚本。
7. 当准备好运行批处理测试时，请单击“运行”按钮。点击“查看结果”按钮，运行完以后可以只查看当前测试任务的测试结果。
8. 可以点击“浏览结果”查看打开测试结果查看器查看所有的测试任务的测试结果。
9. 在测试结果查看器中可以导出测试结果到 XLS 格式文档中。

使用测试报告

查看测试结果

运行测试或组件后，可以查看运行会话期间发生的主要事件的报告。

本章描述：

- 关于查看测试报告
- 了解“测试结果”窗口
- 查看运行会话的结果
- 查看检查点结果
- 查看参数化值和输出值结果
- 删除测试结果

关于查看测试报告

当运行会话结束后，可以在“测试结果”窗口中查看测试结果。默认情况下，“测试结果”窗口在运行结束时自动打开。

“测试结果”窗口包含在运行会话期间执行的步骤的描述。对于组件或不包含数据表参数的测试，“测试结果”窗口显示单个测试循环。

如果测试包含数据表参数，并且将测试设置配置为运行多次循环，则“测试结果”窗口显示测试运行的每次循环的详细信息。结果按测试中的操作分组。

运行测试或组件后，“测试结果”窗口将显示运行会话的所有方面，包括：

- 高级结果概述报告（通过/失败状态）
- 所有运行中使用的数据
- 可展开的步骤树，指明应用程序发生失败的确切位置
- 测试或组件中发生失败的确切位置
- 突出显示测试或组件中失败的步骤
- 测试或组件的每个阶段中，每个步骤和检查点通过或失败的详细说明

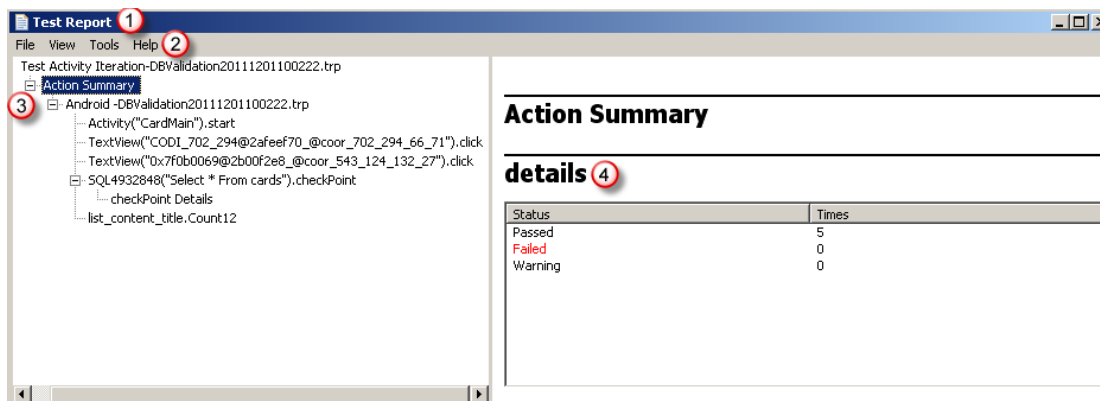
了解“测试结果”窗口

运行会话结束后，可以在“Test Report”窗口中查看结果。默认情况下，运行会话结束时将打开“Test Report”窗口。有关更改默认设置的信息，请参阅第 N 页的“定义测试的运行设置”。

注意：可以从工具栏上，将指定的“Test Report”窗口作为独立的应用程序打开。要打开指定的“Test Report”窗口，请选中测试结果文件 > 单击“Open Test Report”，或选择

“Action” > “Open Test Report”

下例显示了一个对数据库记录校验的测试的结果：



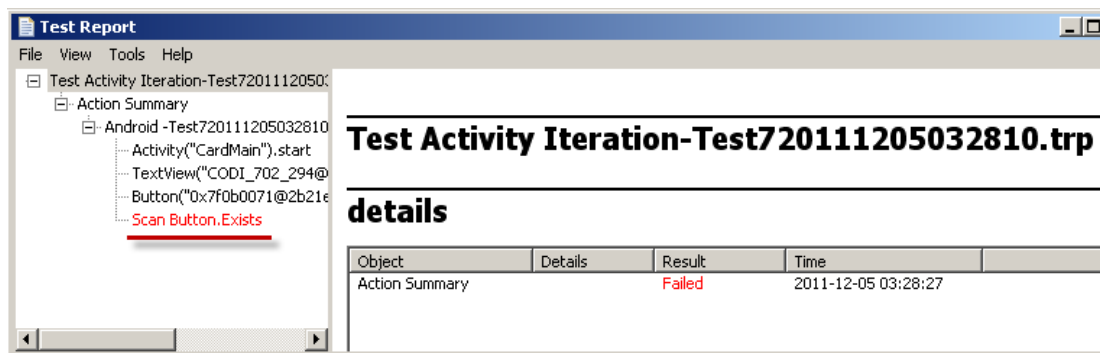
上例显示的测试的结果按操作进行组织。“测试结果”窗口组要包含 4 大部分：

1. 测试结果标题栏 - 显示测试或组件的名称。
2. 测试结果菜单栏 - 显示可用命令的菜单。

测试结果树 - 测试结果的图形表示。有关详细信息，请参阅下面的“测试结果树”。

测试结果详细信息 - 包含选定步骤的详细信息。有关详细信息，请参阅下面的“测试结果详细信息”。

以下是一个包含失败测试结果的示例，高亮显示测试或组件中失败的步骤：



测试结果树

“测试结果”窗口的左窗格中显示测试结果树 - 测试结果的图形表示：

- 黑色字体表示成功的步骤。
 - 红色字体表示失败的步骤。注意，这也会导致所有父步骤（直至根操作或测试）失败。
 - 橙色字体表示意外失败的步骤，例如找不到检查点的对象，将会导致随后的步骤都被阻碍。
- 可以折叠或展开测试结果树中的分支，以便查看该树显示的详细信息的级别。其中指明测试名、系统名、对象名、操作步骤名、结果名。

测试结果详细信息：

默认情况下，打开“测试结果”窗口时，窗口的右侧部分将显示测试或组件的概要。其中指明对象名、操作步骤名、结果名、运行的日期和时间。

在树中选择分支或步骤时，右窗格中将显示选定项目的详细信息。

查看运行会话的结果

默认情况下，在运行会话结束时，将在“测试结果”窗口中显示结果。（可以在“设置”对话

框中更改默认设置。有关详细信息，请参阅第 N 页的“定义测试的运行设置”。)

此外，还可以查看当前测试或组件的以前的运行结果，以及其他测试和组件的结果。还可以在屏幕上预览测试结果，并将它们打印到默认的 Windows 打印机。

要查看运行结果，请执行下列操作：

1. 如果尚未打开“测试结果”窗口，请首先在项目树上展开“测试结果”文件夹，选中测试结果文件 > 单击工具栏上的“打开测试结果”按钮。

提示：可以从“活动”菜单，将“测试结果”窗口作为独立的应用程序打开。

要打开“测试结果”窗口，请选择“Action” > “Open Test Report”。

➤ 如果当前测试或组件具有测试结果，则“测试结果”窗口中将显示这些结果。有关“测试结果”窗口的信息，请参阅第 34 页的“了解“测试结果”窗口”。

➤ 如果当前测试或组件有多个测试结果，或者当前测试或组件没有任何测试结果，则会打开“打开测试结果”对话框。可以选择任何测试或组件的测试结果，或者可以在文件系统中的任何位置搜索测试结果 (results.trp) 文件。单击“打开”可以在“测试结果”窗口中显示选定的结果。

2. 可以折叠或展开测试结果树中的分支，以便浏览报告树显示的详细信息的级别。

➤ 要折叠分支，请选择该分支并单击分支图标左侧的折叠 (-) 符号，或者按数字小键盘上的减号键 (-)。结果树中不再显示分支的详细信息，并且折叠符号变成扩展符号 (+)。

➤ 要折叠测试结果树中的所有分支，请选择“视图” > “全部折叠”或右键单击某个分支并选择“全部折叠”。

➤ 要展开分支，请选择该分支并单击分支图标左侧的展开 (+) 符号，或者按数字小键盘上的加号键 (+)。树中将显示分支的详细信息，并且展开符号变成折叠符号。

如果您刚打开测试结果，则树一次将展开一级。如果树以前是展开的，则将还原为以前的状态。

➤ 要展开某个分支及其下面的所有分支，请选择该分支并按数字小键盘上的星号 (*) 键。

➤ 要展开测试结果树中的所有分支，右键单击某个分支并选择“全部展开”；或者选择树的顶层并按数字小键盘上的星号 (*) 键。

-
3. 选择“文件” > “退出”以关闭“测试结果”窗口。

查看检查点结果

通过向测试或组件中添加检查点，可以将诸如网页、文本字符串、对象属性和表中的预期值与应用程序中这些元素的值进行比较。这样可以确保应用程序的运行符合您的需求。

运行测试或组件时，DroidPilot 将检查点的预期结果与当前结果进行比较。如果结果不匹配，则检查点将失败，这会导致测试或组件失败。可以在“测试结果”窗口中查看检查点的结果。

要查看检查点结果，请执行下列操作：

1. 在“测试结果”窗口中显示测试或组件的测试结果。有关详细信息，请参阅第 35 页的“查看运行会话的结果”。

- 在“测试结果”窗口的左窗格中，展开测试结果树的分支，然后单击要查看其结果的检查点的分支。“测试结果”窗口中将显示检查点结果。

查看参数化值和输出值结果

可以在“测试结果”窗口中查看有关参数化值和输出值步骤的结果的信息。还可以查看运行时数据表的内容。

在测试结果窗口中查看参数化值

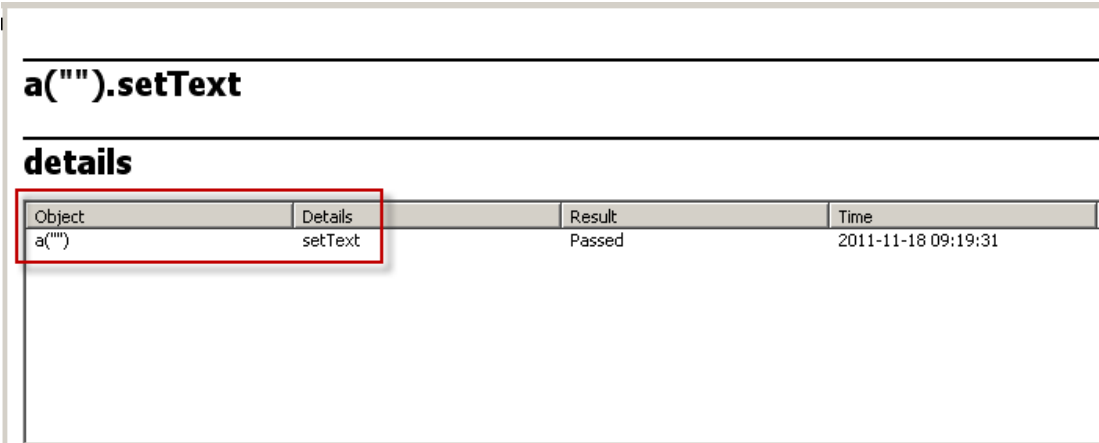
参数是从外部数据源或生成器为其赋值的变量。有关在测试和组件中定义和使用参数的详细信息，请参阅第 12 章“参数化值”。

可以在“测试结果”窗口中查看测试或组件中定义的参数的值。

要查看参数化值，请执行下列操作：

- 在“测试结果”窗口中显示测试或组件的测试结果。有关详细信息，请参阅第 35 页的“查看运行会话的结果”。
- 在“测试结果”窗口的左窗格中，展开测试结果树的分支，然后单击包含参数化值的测试、操作、组件或步骤的分支。

右下窗格中将显示输入参数的名称和值。



The screenshot shows the 'Test Results' window. At the top, the test step is identified as 'a("").setText'. Below this, a section titled 'details' contains a table with four columns: 'Object', 'Details', 'Result', and 'Time'. The first row of data shows 'a("")' in the 'Object' column, 'setText' in the 'Details' column, 'Passed' in the 'Result' column, and '2011-11-18 09:19:31' in the 'Time' column. A red rectangle highlights the first two columns of this row.

Object	Details	Result	Time
a("")	setText	Passed	2011-11-18 09:19:31

上面的示例显示了为操作定义的数据表参数“ParameterTest”，该数据表参数循环的每一个值都将显示出来。

有关在测试和组件中定义和使用参数的详细信息，请参阅“参数化值”。

在“测试结果”窗口中查看输出值结果

这里输出值是一个步骤，在运行会话期间的该步骤中将输出人为期望打印出来的信息。通常这种输出信息用于逻辑功能判断语句，根据运行会话期间得出的实际流程走向输出期望看到的信息到测试报告里。

下列中对象名称和详细信息都是脚本里人为期望输出的信息。

Scan Button.Non-Exists

details

Object	Details	Result	Time
Scan Button	Non-Exists	Failed	2011-12-05 02:44:04

对于测试结果为“失败的”的会有针对性的截图，标注在哪个测试步骤失败。

删除测试结果

使用测试结果删除工具可以按照您定义的特定条件，从系统中删除不必要的或废弃的测试结果。这样可以释放宝贵的磁盘空间。

此工具提供了 Windows 风格的用户界面，在项目树上的“测试结果”节点下，显示了所有测试结果的列表。然后，可以删除不再需要的任何测试结果。

使用测试结果删除工具，可以按名称、日期、大小等等对测试结果进行排序，以便轻松地标识要删除的结果。

要使用测试结果删除工具删除测试结果，请执行下列操作：

1. 展开项目树上的“测试结果”节点。
2. 选中希望删除的结果文件夹，或单个测试结果文件 > 点击鼠标右键 > “删除”

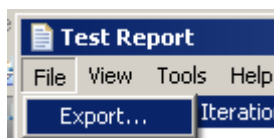
注意：此处将会物理删除整个文件从磁盘路径。

还可以使用如下方式浏览到测试或文件夹：

- 打开本地测试项目路径文件夹
- 展开到指定的测试结果文件夹，可以浏览，或删除所有的测试结果文件。

导出测试报告

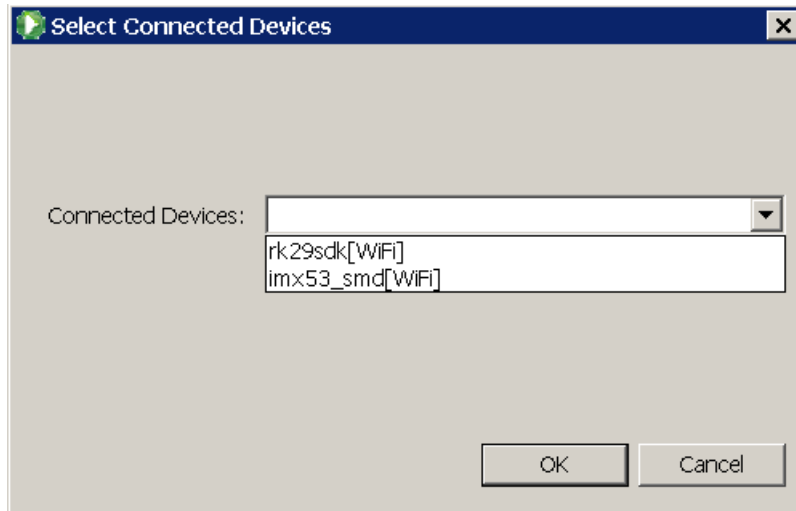
当运行会话结束后，可以在弹出的“测试结果”窗口将测试结果导出到指定的文件格式中，如 Excel，Word 等。



要导出测试结果，请执行下列操作：

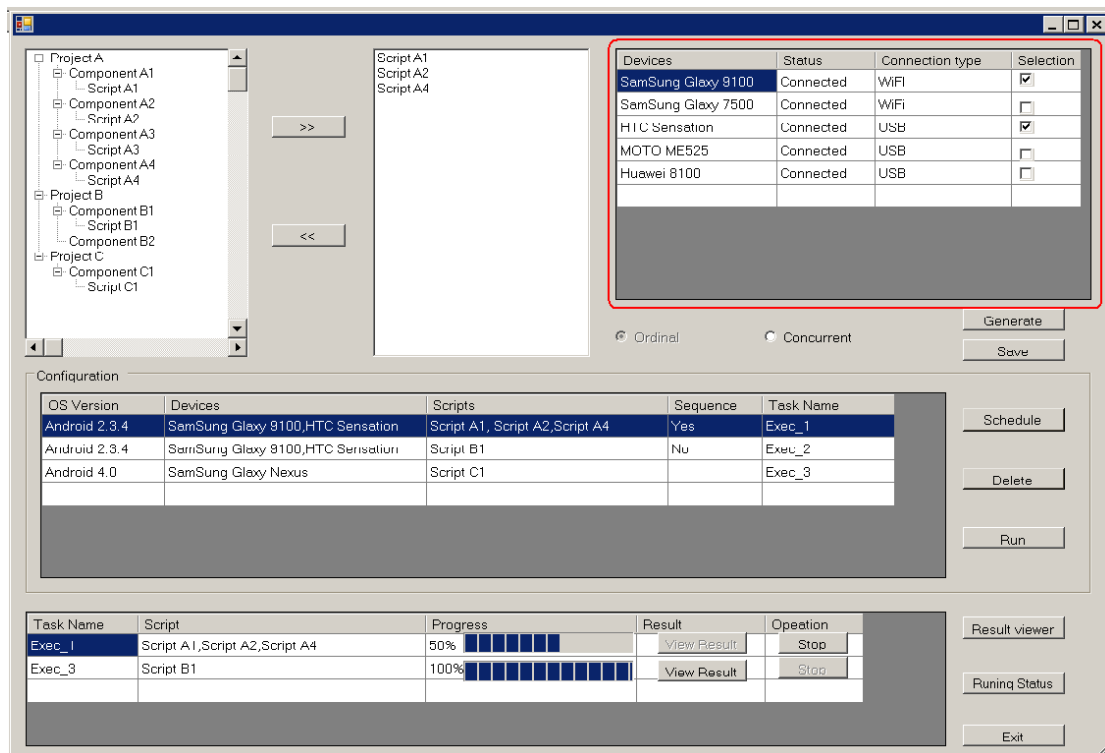
1. 单击“文件”菜单。

2. 单击“导出”
3. 定义“文件标题”
4. 选择“文件类型”
5. 单击 OK，选择保存测试报告的路径。



多设备管理

DroidPilot 的多设备管理界面使用上与单设备没有区别。只是在回放脚本时候会列出所有连接的设备供用户勾选。有关详细信息，请参阅 [“运行批处理测试”](#)



第 VI 部分 使用支持 DroidPilot 环境

使用应用部署工具

DroidPilot 具有自动安装部署功能，可以通过 DroidPilot 应用部署程序进行自动的安装被测应用、DroidPilot 代理程序。

本章描述：

- 使用部署工具安装应用与代理程序
- 部署在多台移动设备

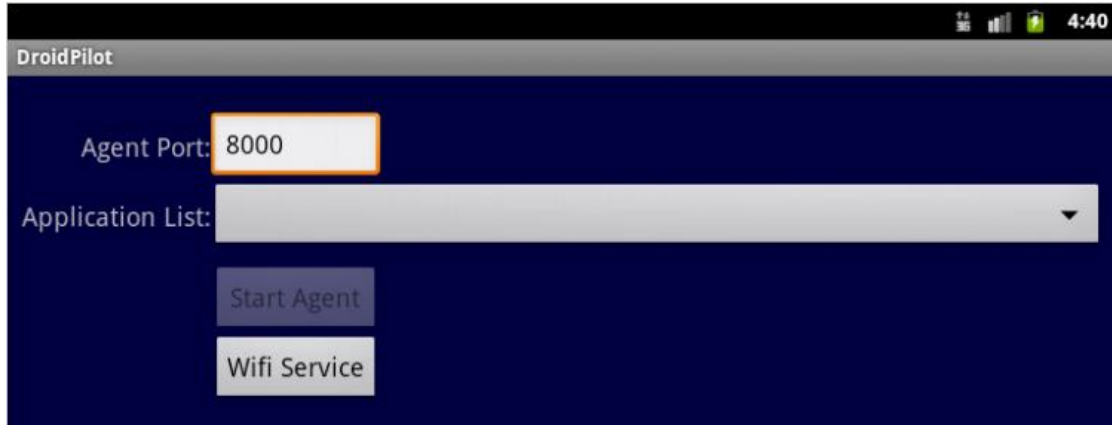
使用部署工具安装应用于代理程序

- 安装代理程序到移动设备
- 安装被测应用到移动设备

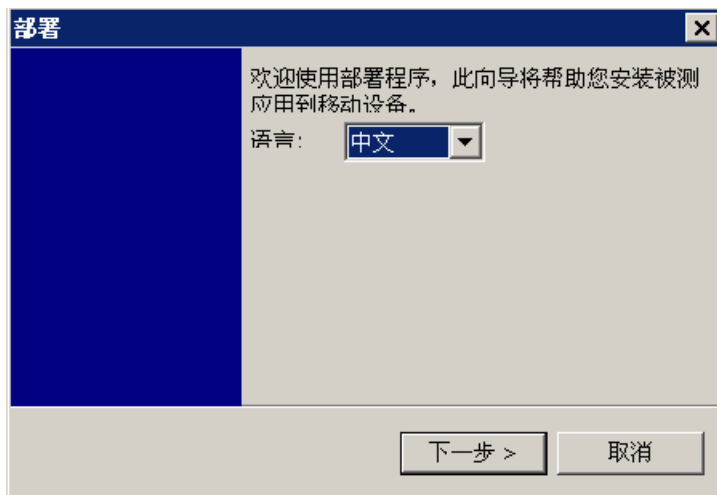
可以安装一个 DroidPilot 部署工具用以自动安装被测应用和部署 DroidPilot 代理程序。您可以通过一个方便的向导方式来完成对应用的部署。

首先您需要安装代理程式到移动设备，并运行它，DroidPilot 的代理程式是一个 APK 的档，需要您通过 ADB 命令来安装（预先安装 USB 驱动程序，或者通过类似安装应用的协力厂商工具来安装，比如“豌豆荚”“手机 91 助手”）。该 APK 只需要在第一次运行的时候安装。如果设备上已经安装了该 APK，无需重复安装。





通過部署工具安裝被測應用到移動設備



您可以按照向导一步步的安装您的应用以及 DroidPilot 的代理程序到移动设备中，有兩種安裝連接方式供您選擇：USB 以及 Wifi 的方式：



USB 的連接方式：表示移動設備和您的電腦連接通過 USB 線的方式連接，請與預先安裝好該移動設備的 USB 驅動程式，確保移動設備通過 USB 正常的連接到您的電腦。

WiFi 的連接方式：表示您的電腦與移動設備通過 WiFi 連接到一個局域網內，會要求輸入 IP 位址與埠號

如何獲取移動設備的 IP 位址與埠號：請預先在移動設備上啟動 Agent 程式中的“WiFi 服務”

一旦完成以上操作步骤，您要做的就是移动设备端通过代理程序启动被测应用。

一旦您完成应用的安装和代理程序的部署，会在 DroidPilot 的代理程序的应用启动列表中出现你的应用。

您只需要输入该应用的启动的端口号用以支持 DroidPilot 客户端的设备管理模块中对该设备的连接。

部署在多台移动设备

DroidPilot 的部署工具同时还考虑到用户将会安装应用到批量设备上去，一次性自动安装多个应用到多台移动设备上去。用户按住 CTRL 键多选应用可以安装多个 APK 应用到移动设备中

第 VII 部分 附录

附录 I - DroidPilot 安装指南

DroidPilot 安装包含 DroidPilot Designer, DroidPilot Agent 的安装, 其中 DroidPilot agent 的安装需要通过 DroidPilot 的 Deployment tool 安装, 因此安装包含 DroidPilot designer, DroidPilot Deployment 两组件的安装.

本章描述:

- 安装之前准备
- 安装 DroidPilot Designer

安装之前

欢迎使用 AndroidPilot 移动应用测试解决方案 - DroidPilot Mobile Automation Testing。
本指南讲述在独立计算机上安装 DroidPilot 所必备的知识。

在安装 DroidPilot 之前, 请检查系统要求。

系统要求

要成功安装并运行 DroidPilot, 您需要满足以下最低系统要求:

计算机/处理器 IBM-PC 中央处理器 Intel Core i3 1.2MHZ 或以上更快的微处理器兼容的计算机。

操作系统 Windows 2000 - SP4、Windows XP - SP3 或者 Windows 7 SP1 及以上, Windows 2003 Server。

注意: 只有未激活执行防止机制 (又称为 DEP - 数据执行防止和 NX - 无执行) 时, 才能在 Windows XP SP3 上支持 DroidPilot。

内存至少 2GB 的内存, 计算机内存在 4GB 或以上 DroidPilot 会运行的更加流畅。

可用磁盘空间不低于 2048MB 可用磁盘空间用于应用程序文件和文件夹, 另外在系统盘 (安装操作系统的磁盘) 上需要 120 MB 可用磁盘空间。安装 DroidPilot 后, 建议系统盘中至少留有 150 MB 可用磁盘空间, 以便操作系统和 DroidPilot 正确运行。

JDK: 运行 DroidPilot designer 需要 Java JRE, 至少需要 Java™ SE Runtime Environment Build 1.6.0 以上版本并设置 Java 开发环境变量。

最新下载安装 URL: http://www.java.com/zh_CN/

安装 DroidPilot

要安装 DroidPilot，必须知道要运行的安装类型（典型还是自定义），以及安装的许可证方式（试用版，单机版，商业版）。

本章描述：

- 了解不同许可证类型的 DroidPilot 安装
- 安装 DroidPilot
- 不同许可证版本的功能说明

了解不同许可证类型的 DroidPilot 安装

对于“试用”许可证安装和“商业”许可证安装，安装 DroidPilot 和激活许可证的常规过程不同。也可安装 DroidPilot 试用版许可证，该许可证可使用 15 天。本部分概述了针对这两种许可证类型的各个过程，您可以直接浏览每一步的详细信息。

安装 DroidPilot

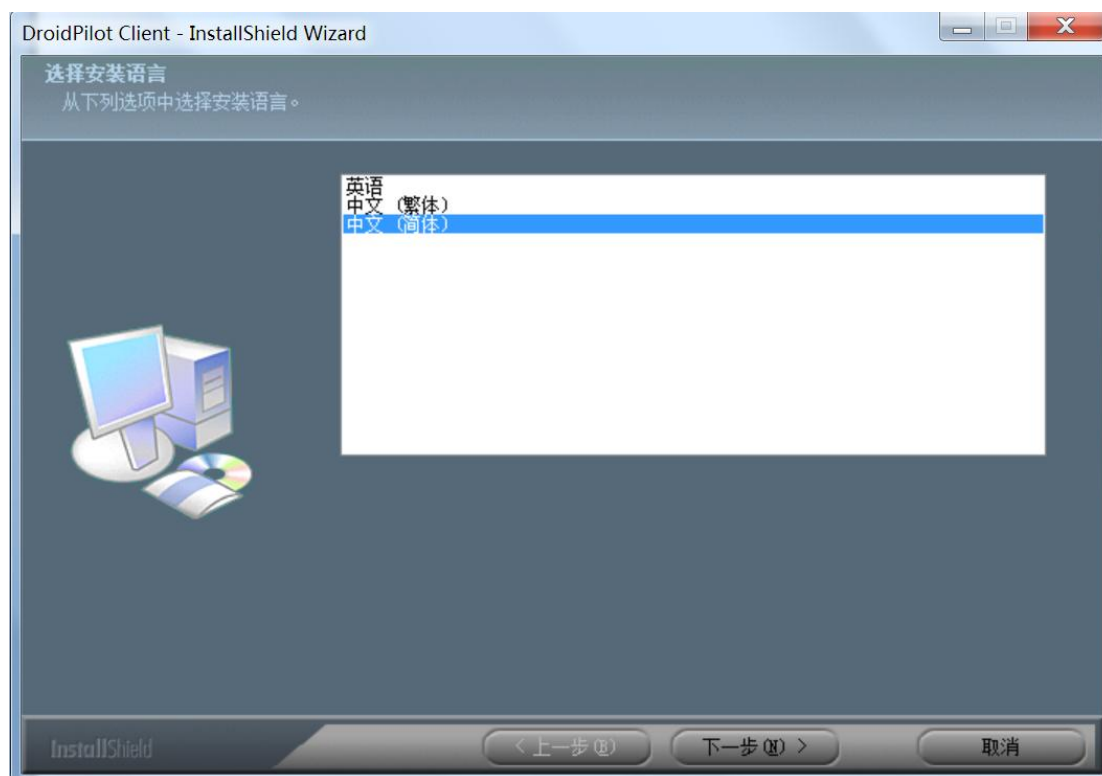
DroidPilot 会采用安装向导方式一步步指导你安装

使用试用单机版许可证安装 DroidPilot

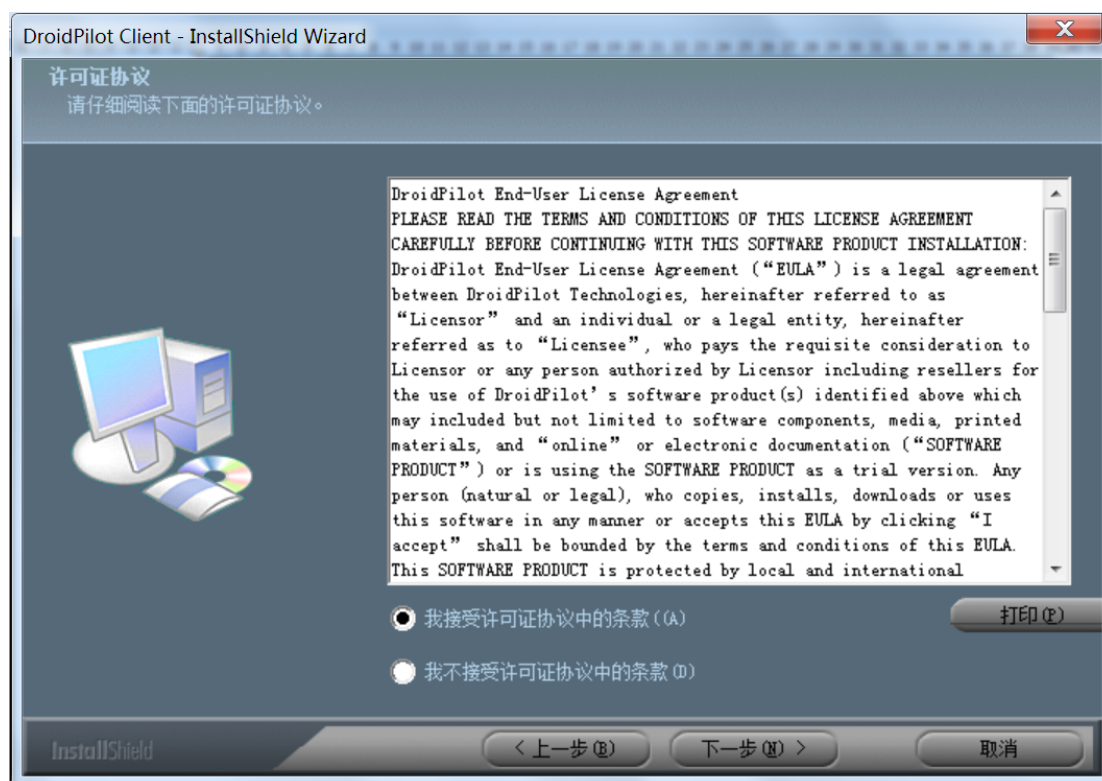
本部分概述使用试用单机许可证安装 DroidPilot 的过程。

要使用试用许可证安装 DroidPilot，请执行下列操作：

1. 运行 DroidPilot 安装程序，选择一种安装语言



2. 点击“接受许可证协议中的条款”

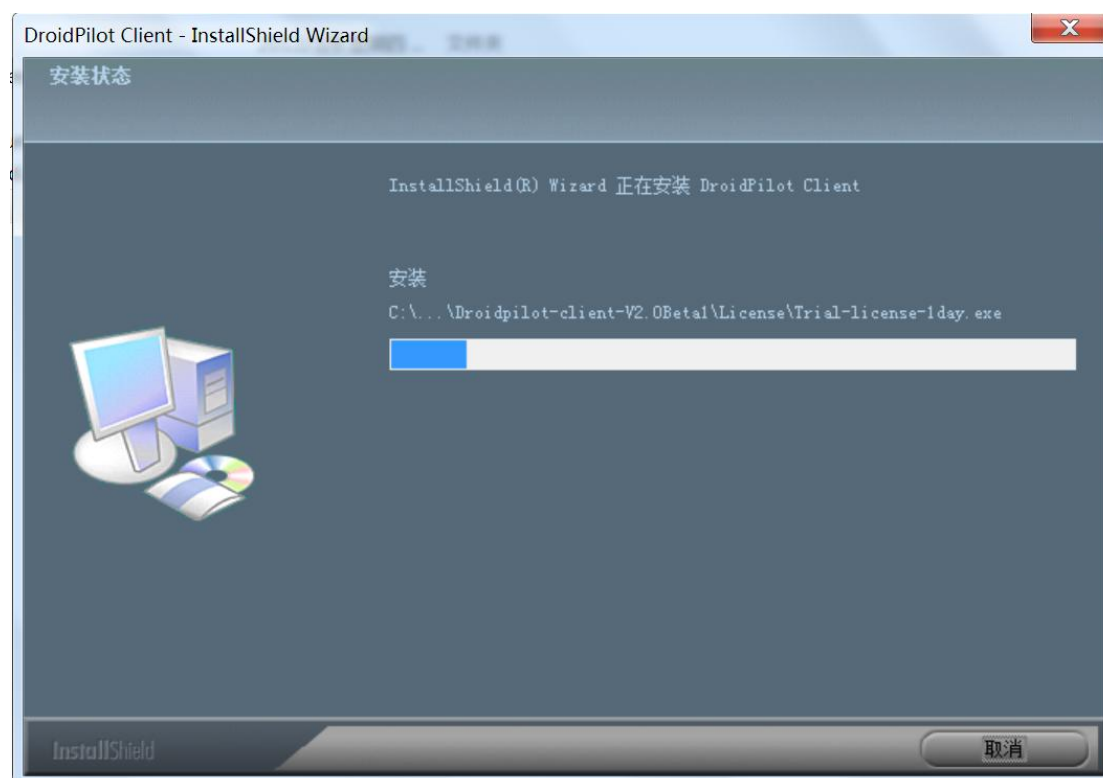


3. 输入用户名公司名后，选择安装类型（完全还是定制）



一般都选完全，完全安装包含 DroidPilot Designer， DroidPilot Deployment tool， 以及 Documents 三部分，其中 Designer 与 DroidPilot Deployment tool 必装，如果再次重装可以选择定制安装，比如只需要安装 DroidPilot Designer

4. 点击“下一步”后开始复制档



5. 系统自动安装许可证验证程序的驱动，同时也完成对试用版的许可证文件的导入，该许可证

允许用户免费使用 15 天。

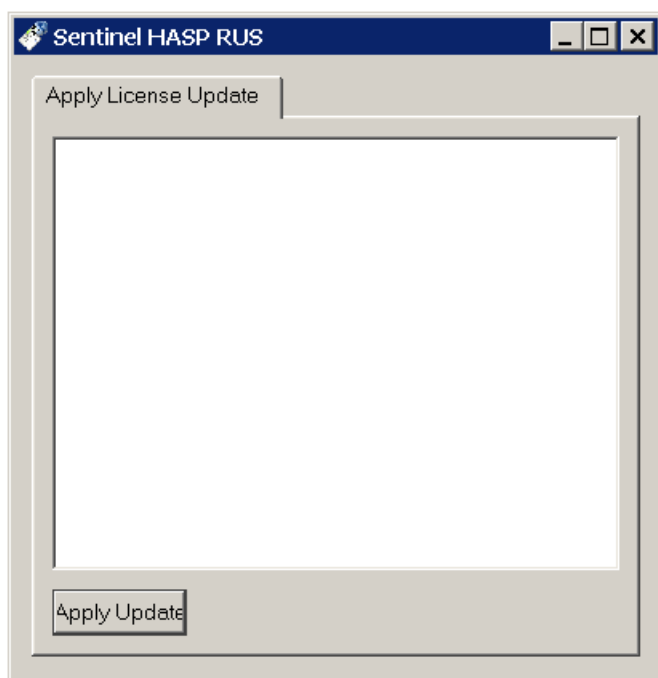
6. 点击“完成” 完成 后在开始菜单，以及桌面上中可以看到安装好的 DroidPilot 组件快捷图标



使用正式单机版许可证安装 DroidPilot

正式的单机版安装与试用版的安装基本相同，不同之处在许可证文件的导入需要使用者自行用购买的许可证文件导入，许可证验证驱动安装同试用版相同，详细请参阅试用版安装步骤 5

用户执行购买的许可证导入程序点击“Apply Update”按钮导入许可证文件。请确保在 DroidPilot 安装程序完成以后使用。如之前有过安装许可证验证驱动程序，可无须再次安装



不同许可证版本的功能说明

产品特性	试用版	商业版	企业版
并发用户数量	一个用户	一个用户	一个用户
使用期限	15 天	3/6/12 个月	3/6/12 个月/永久
技术支持	邮件	邮件+电话+专业服务*	邮件+电话+专业服务
连接设备数量	一台	一台	多台（含 AutoRunner）
功能	全部功能除支持多台设备	全部功能除支持多台设备	全功能
运行在虚拟机	不可	可以	可以
测试脚本行数限制	有行数限制	无限制	无限制
整合测试管理平台	无	可整合*	可整合

关于专业服务与测试管理平台的说明：

1. 仅支持商业版
2. 服务内容包括：培训及顾问服务，运行环境搭建及维护，测试脚本设计与开发。

附录 II - 了解基本的 VBScript 语法

1. 条件语句，用于判断以及分支操作

If ..Then ..[Elseif]..[Else]..End If

2. GetROProperty 方法:

获取当前对象的某一个属性值，一般用于条件语句内

3. Reporter.ReportEvent:

在 Test Report 内打印正确和错误记录

打印正确报告 Reporter.ReportEvent micPass,String1,String2

打印错误报告 Reporter.ReportEvent micFail,String1,String2

Script Example:

```
If Window("MainActivity-3").TextView("list_content_title").GetROProperty("Text")="共[count]张名片" Then
    Reporter.ReportEvent micPass,"The count is correct","This message is user defined"
Else
    Reporter.ReportEvent micFail,"The count is incorrect","This message is user defined"
End If
```

4. 标准检查点:

用于检查 PlayBack 过程中某个对象的属性值是否与预期的属性值一样,运行后会在 Test Report 中记录结果

```
Window().Button().checkPoint=<(PropertyName1, ExpectedValue); (PropertyName2, ExpectedValue)>
```

Script Example:

```
Window("MultiGroup").Button("android.widget.Button@2b17e270").checkPoint=<(isEnabled,false)>
```

5. 数据库检查点:

通过连接数据库，利用 SQL 查询语句用来检查一些业务数据的正确性，一般将返回值用在条件语句内

database(dbPath="?",pkgName="?",dbName="?",sql="?",args="?",return="?")依次输入数据库在移动应用中的路径，包名，数据库名，sql 语句，参数，返回的列名

Script Example:

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz.cardname",dbName="card.db",sql="Select count(*) as countname From cards",args="",return="countname")
```

//返回 cards 这个表的总数量。返回的值可以用[]括起来当成字符串用

```
If "总共[countname]张名片"
```



```
=Window("FamilyName2").TextView("android.widget.TextView@2b12b530").GetROProperty("Text") Then
    Reporter.ReportEvent micPass,"SearchRecords","Search by 'A' correctly"
Else
    Reporter.ReportEvent micFail,"SearchRecords","Search by 'A' Incorrectly"
End If
```

当数据库查询有多个参数时，格式如下：

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz
.cardname",dbName="card.db",sql="Select name as countname From cards Where head like '%
杨%' or spell like '%Y%'",args="",return="countname")
```

或是：

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz
.cardname",dbName="card.db",sql="Select name as countname From cards Where head like ? or
speel like ?",args="%杨%/%Y%",return="countname")
```

6. sleep(microsecond)方法

Script Example:

```
sleep(3000) //休眠 3 秒
```

7. Exist 方法:

判断某个 Object 在操作到的当前页面是否存在，一般用于条件语句中

Script Example:

```
If Window("FamilyNames").TextView("android.widget.TextView@2b129650").Exist=true Then
    Reporter.ReportEvent micPass,"get result","successfully"
Else
    Reporter.ReportEvent micFail,"get result","Fail"
End If
```

8. 支持 contains, in 比较符号，书写格式如下:

a contains b （当 a 包含了 b 时，返回 true, 否则 false)

a in b （当 b 包含了 a 时，返回 true, 否则 false)

Script Example:

a.(contains):

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz
.cardname",dbName="card.db",sql="Select name as countname From cards Where head like '%
杨%' or spell like '%Y%'",args="",return="countname")
```

If "[countname]" contains "杨杰" Then

```
    Reporter.ReportEvent micPass,"SearchRecords","Search correctly"
Else
    Reporter.ReportEvent micFail,"SearchRecords","Search Incorrectly"
End If
```

b.(in):

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz
.cardname",dbName="card.db",sql="Select name as countname From cards Where head like '%
杨%' or spell like '%Y%'",args="",return="countname")
```

```
If "杨杰" in "[countname]" Then
```

```
Reporter.ReportEvent micPass,"SearchRecords","Search correctly"
```

```
Else
```

```
Reporter.ReportEvent micFail,"SearchRecords", "Search Incorrectly"
```

```
End If
```

9. 支持=, ==, >=, >, <=, < 比较符号, 跟数值进行比较, 格式如下:

```
a = b
```

```
a == b
```

```
a >= 12
```

```
a > 12
```

```
a <= 12
```

```
a < 12
```

Script Example:

```
database(dbPath="mnt/sdcard/AppFiles/com.ebensz.cardname/card.db",pkgName="com.ebensz
.cardname",dbName="card.db",sql="Select count(*) as countname From cards
",args="",return="countname")
```

```
If "[countname]" == 12 Then
```

```
Reporter.ReportEvent micPass,"SearchRecords","Search correctly"
```

```
Else
```

```
Reporter.ReportEvent micFail,"SearchRecords", "Search Incorrectly"
```

```
End If
```

10. 支持 and, or (或&& ||) 操作符, 格式如下:

```
a == b and c == d or e == f
```

也可以写成

```
a == b && c == d || e == f
```

如下写成 `a == b && (c == d || e == f)`带括号的形式, 则会先对 `c == d || e == f` 进行比较, 再与 `a==b` 比较

11. 支持 longClick, 长按某个应用出菜单

Script Example:

```
Window("HomePage").Button("android.widget.Button@2afee6d8").click
```

```
Window("FamilyNames").ImageView("android.widget.ImageView@2b1ab7a0").longClick
```

12. 支持功能按键

KeyClick.Back_Key: 返回到前一页
KeyClick.Home_Key: 返回到主页
KeyClick.Volume_Up: 声音调大
KeyClick.Soft_Left: 左软键
KeyClick.Soft_Right: 右软键
KeyClick.Menu_Key: 右键菜单
KeyClick.Volume_Down: 声音调小
KeyClick.Media_Play_Pause: 视频暂停播放
KeyClick.Media_Stop: 视频停止播放
KeyClick.Media_Next: 播放下一个视频
KeyClick.Media_Previous: 播放上一个视频
KeyClick.Media_Rewind: 视频倒放
KeyClick.Media_Fast_Forward: 视频快进播放

13. 支持 updateTime: 更新时间

Script Example:

```
Window("MainActivity-1").TimePicker("android.widget.TimePicker@41130a28").updateTime="14:55:44"
```

14. 支持 updateDate :更新日期

Script Example:

```
Window("MainActivity-3").DatePicker("android.widget.DatePicker@411862b0").updateDate="2012-4-26"
```

15. 支持 go on （部分新闻客户端,媒体报纸应用）

在 Android 4.0 的应用，需要在 script 首行添加 go on 语句

16. 其他一些需要注意的点

- a:标准检查点不要坐标 coordinate()
- b:EditText 对象必须加 ActionType("input")或 ActionType("append")或 ActionType("insert")
- c:If 语句内的等于号必须左右无空格
- d:If Else End If 必须注意首字母大写，其他字母小写
- e:写脚本尽量用 script 的助写功能，打个点就可以从下拉列表选取

-- The End --