

Silk Test 17.5

关键字驱动测试

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

Copyright © Micro Focus 1992-2016. All rights reserved.

MICRO FOCUS, the Micro Focus logo and Silk Test are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

All other marks are the property of their respective owners.

2016-11-03

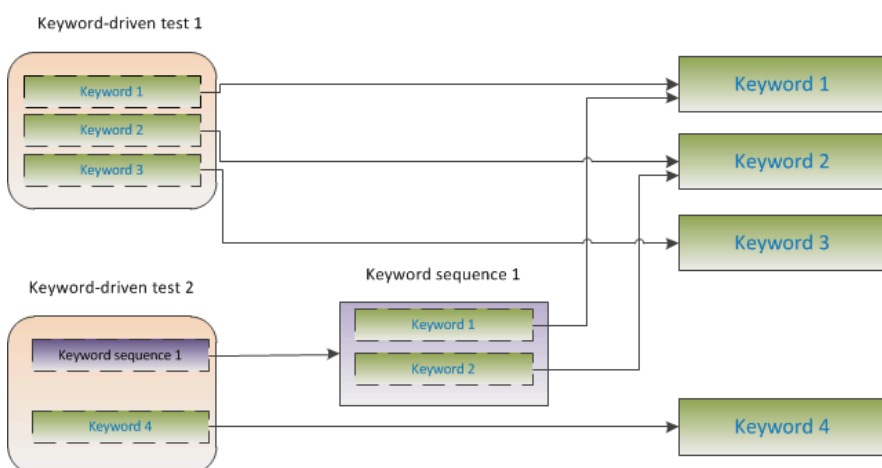
内容

关键字驱动测试	4
测试自动概述	4
关键字驱动的测试的优势	4
关键字	5
将 Silk Test 与 Silk Central 相集成	6
将关键字库上载到 Silk Central	6
通过命令行将关键字库上载到 Silk Central	8
通过自动化手动测试创建关键字驱动的测试	9
在 Silk Central 中创建关键字驱动的测试	9
在 Silk Central 内的测试中管理关键字	10
Silk Test 推荐哪些关键字？	11
使用关键字的参数	12
示例：带参数的关键字	12
在 Silk Test 中创建关键字驱动的测试	14
在 Silk Test 中录制关键字驱动的测试	14
在 Silk Test 中为关键字驱动的测试设置基态。	15
在 Silk Test 中实施关键字	15
在 Silk Test 中实施 Silk Central 关键字	16
在 Silk Test 中录制关键字	16
编辑关键字驱动的测试	17
将关键字组合为关键字序列	17
从 Eclipse 回放关键字驱动的测试	18
从 Silk Central 回放 Silk Test 测试	18
通过命令行回放关键字驱动的测试	19
用 Apache Ant 回放关键字驱动的测试	20
使用特定变量回放关键字驱动的测试	22
分组关键字	22

关键字驱动的测试


关键字驱动的测试是一种软件测试方法，它将测试设计与测试开发分离开来，因此允许其他专业小组参与测试自动化流程，例如业务分析师。Silk Central 和 Silk Test 支持关键字驱动的测试方法，让自动化工程师能开发出由共享资产（采用 Silk Test 中的关键字形式）组成、易于维护的自动化框架，从而支持业务分析师与工程师之间开展密切协作。随后，业务分析师在 Silk Test 中使用这些关键字新建关键字驱动的测试，或者在 Silk Central 中利用这些关键字将其现有手动测试资产转换为自动化测试，或是新建关键字驱动的测试。

- 关键字驱动的测试就是一个关键字序列。关键字驱动的测试可像其他任何测试一样回放。
- 关键字序列就是由其他关键字组合而成的关键字。关键字序列通常将遇到的关键字组合捆绑为一个关键字，这有助于您减少维护工作量并确保测试组织有序。
- 关键字就是针对一个测试对象的一项或多项操作的既定组合。关键字可以使用多种工具和编程语言实施，例如 Java 或 .NET。



关键字驱动的测试的创建需要两个阶段：

1. 设计测试。
2. 实施关键字。

 **注：**以下主题介绍了如何利用 Silk Central 自动化手动测试，以及如何利用 Silk Central 和 Silk Test 执行关键字驱动的测试。使用 Silk Test 执行的任何任务均以 Silk4J (Silk Test 的 Eclipse 插件) 为依据介绍。Silk4NET 和 Silk Test Workbench 的步骤基本类似。有关使用特定 Silk Test 客户端执行关键字驱动的测试的其他信息，请参阅 Silk Test 客户端的文档。

测试自动概述

测试自动化是通过使用验证来比较实际执行结果与预期结果，从而利用软件控制测试执行的过程。在 Silk Test 上下文中，测试自动化包括自动化功能测试和回归测试。自动化功能测试和回归测试让您节省资金、时间和资源。

测试自动化框架是一种设置特定产品的自动化规则的集成化系统。该系统集成了测试、测试数据源、对象详细信息和各种可重复使用的模块。这些组件需要装配在一起才能代表业务流程。框架提供测试自动化的基础，可简化自动化工作。

关键字驱动的测试的优势

使用关键字驱动的测试方法的优势如下：

- 关键字驱动测试会将测试自动化与测试用例设计分离开来，从而支持技术人员（实施关键字的测试工程师）与非技术人员（设计测试用例的主题专家）之间的更好分工合作。
- 可以提前开发测试，无需访问所测试的应用程序，关键字可以稍后再予实施。
- 不具备编程知识也能开发测试。
- 从长期角度来看，关键字驱动的测试所需维护更少。您需要维护关键字，所有使用这些关键字的关键字驱动的测试都将自动更新。
- 测试用例简明扼要。
- 测试用例更易于非技术受众阅读和理解。
- 测试用例易于修改。
- 新测试用例可以重复使用现有关键字，这种做法的优势之一就是更容易提高测试覆盖率。
- 关键字实施的内在复杂性对于需要创建或执行关键字驱动的测试的用户而言不可见。

关键字

关键字就是针对一个测试对象的一项或多项操作的既定组合。关键字可以使用多种工具和编程语言实施，例如 Java 或 .NET。在 Silk Test 中，关键字是带批注的测试方法 (@Keyword)。关键字将以关键字资产的形式保存。

您可以在创建关键字驱动的测试期间定义关键字和关键字序列，随后可以将其实施为测试方法。还可以用 @Keyword 批注将现有测试方法标记为关键字。在 Java 中，关键字是使用以下批注定义的：


```
@Keyword("keyword_name")
```

关键字序列就是由其他关键字组合而成的关键字。关键字序列通常将遇到的关键字组合捆绑为一个关键字，这有助于您减少维护工作量并确保测试组织有序。

一个关键字可以具有输入和输出参数。实施关键字的测试方法的任意参数都是关键字的参数。要为关键字的参数指定不同的名称，可以使用以下：

```
// Java code
@Argument("parameter_name")
```

默认情况下，Silk4J 中的参数就是指输入参数。若要定义输出参数，请使用 OutParameter 类。

 **注：**要在关键字驱动的测试编辑器中为关键字指定输出参数，请使用以下批注：

```
${parameter_name}
```

在关键字驱动的测试编辑器中，可以使用相同的批注，将一个关键字的输出参数用作其他关键字的输入参数。

示例

标记为关键字的测试方法可能如下所示：

```
// Java code
@Keyword("Login")
public void login(){
    ... // method implementation
}
```

或

```
// Java code
@Keyword(value="Login", description="Logs in with the given
name and password.")
public void login(@Argument("UserName") String userName,
    @Argument("Password") String password,
    @Argument("Success") OutParameter success) {
    ... // method implementation
}
```

关键字将使用给定用户名和密码登录到应用程序，并返回登录是否成功。要将输出参数用作其他关键字的输入参数，需要在关键字内设置输出参数的值。



注: 如果以 PDF 格式查看本帮助，此代码示例可能包含换行符，脚本不允许换行符。若要在脚本中使用此代码示例，请去掉这些换行符。

- **Keyword** 批注的关键字名称参数为可选。您可以使用关键字名称参数指定与方法名称不同的名称。如果未指定该参数，则方法名称将用作关键字名称。
- **Argument** 批注也是可选的。如果一个方法标记为关键字，则其所有参数都将自动用作关键字参数。您可以使用 **Argument** 批注为关键字参数指定不同的名称，例如，指定 *UserName* 而非 *userName*。

将 Silk Test 与 Silk Central 相集成

将 Silk Test 与 Silk Central 相集成，以支持技术用户和技术水平较低的用户之间的协作。

在 Silk Test 与 Silk Central 相互集成，并且 Silk Central 中存在与活动 Silk Test 项目同名的库时，**Silk4J > 显示关键字视图**下的**关键字**视图除了显示活动 Silk Test 项目中定义的任何关键字之外，还会显示 Silk Central 库中的所有关键字。



注: Silk Central 连接信息为每个 Silk Test 用户单独存储，这意味着要使用 Silk Central 中的关键字和关键字序列的每个 Silk Test 用户必须将 Silk Test 与 Silk Central 集成。

将 Silk Test 与 Silk Central 相集成可为您带来如下优势：

- 测试管理和执行由 Silk Central 处理。
- 关键字存储在 Silk Central 数据库（上载库）内，可供 Silk Central 中的所有项目使用。
- 手动测试可以直接在 Silk Central 中自动执行，创建的关键字驱动的测试可以从 Silk Central 在 Silk Test 中执行。



注: 在 Silk Test 中，可以编辑和执行位于 Silk Test 中的关键字驱动的测试，并且可以执行存储在 Silk Central 中的关键字驱动的测试。若要编辑存储在 Silk Central 中的关键字驱动的测试，请在**关键字驱动的测试编辑器**中打开关键字驱动的测试，然后单击**编辑**。

以下步骤展示了如何将 Silk4J 与 Silk Central 相集成。Silk4NET 和 Silk Test Workbench 的步骤基本类似。有关使用特定 Silk Test 客户端执行关键字驱动的测试的其他信息，请参阅 Silk Test 客户端的文档。

1. 在菜单中，选择 **Silk4J > Silk Central 配置**。将打开**首选项**对话框。
2. 在 **URL** 字段中键入您的 Silk Central 服务器的 URL。
例如，如果 Silk Central 服务器名称为 *sctm-server*，并且 Silk Central 的端口为 13450，请键入 `http://sctm-server:13450`。
3. 在相应字段中键入有效用户名和密码。
4. 单击**验证**以验证 Silk Test 能否通过指定用户访问 Silk Central 服务器。
5. 单击**确定**。

将关键字库上载到 Silk Central

要使用 Silk Central，请确保您已配置有效的 Silk Central 位置。有关其他信息，请参阅[将 Silk Test 与 Silk Central 相集成](#)。

要在 Silk Central 中自动化手动测试，请将您已在 Silk Test 项目中实施的关键字以关键字库的形式上载到 Silk Central，随后即可从中利用关键字自动化手动测试。

1. 在 Silk Test 中，选择包含关键字驱动的测试的项目。

2. 确保 Silk Central (测试 > 库) 中存在相同名称的库。
3. 在工具栏中, 单击**上载关键字库**。
4. *可选*: 提供关键字库更改的说明。
5. *可选*: 单击**配置**来配置与 Silk Central 的连接。
6. *可选*: 若要查看连接的 Silk Central 实例中有哪些库可用, 请单击链接。
7. 单击**上载**。



警告: 如果 Silk Central 中的关键字库已经分配给其他自动化工具或另一个 Silk Test 客户端, 那么系统会询问您是否确实要更改关键字库的类型。只有在确定要更改类型的情况下才上载库。

Silk Test 将使用项目中已实施的所有关键字创建一个关键字库。随后, Silk Test 使用名称 library.zip 将关键字库保存到项目的输出文件夹中。系统会验证库, 以保证一致性, 可能破坏 Silk Central 中现有测试的任何更改将在**上载关键字库至 Silk Central** 对话框中列出。最后, Silk Test 会将该库上载到 Silk Central。现在, 您可以在 Silk Central 中使用关键字。Silk Central 中任何使用关键字库内所含关键字的关键字驱动的测试都会自动使用关键字的当前实施。

从 Silk Test 15.5 中创建的项目上载关键字库

要从使用 Silk Test 15.5 创建的 Silk Test 项目上载关键字库, 您需要编辑项目的 build.xml 文件。

1. 在**程序包管理器**中, 展开要从中上载关键字库的项目的文件夹。
2. 打开 build.xml 文件。
3. 将项目的关键字资产目录添加到**编译**目标的 JAR 构建步骤:

```
<fileset dir="Keyword Assets" includes="**/*.kwd"
erroronmissingdir="false" />
```

4. 为关键字库添加以下目标:

```
<target name="build.keyword.library" depends="compile">
  <java
    classname="com.borland.silk.kwd.library.docbuilder.DocBuilder"
    fork="true">
    <classpath refid="project.classpath" />

    <arg value="AutoQuote Silk4J Library" />
    <arg value="{output}" />
    <arg value="{output}/library.zip" />
  </java>
</target>
```

新 build.xml 文件应与以下类似:

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="AutoQuote" default="compile">

  <property name="src" value="src" />
  <property name="bin" value="build" />
  <property name="output" value="output" />
  <property name="lib" value="lib" />
  <property name="builddlib" value="builddlib" />

  <path id="project.classpath">
    <fileset dir="{lib}" includes="*.jar"
excludes="*source*" />
    <fileset dir="{builddlib}" includes="*.jar"
excludes="*source*" />
  </path>

  <target name="clean">
    <delete dir="{output}" />
  </target>
```

```

<target name="compile" depends="clean">
  <mkdir dir="${output}" />

  <delete dir="${bin}" />
  <mkdir dir="${bin}" />

  <componentdef name="ecj"
  classname="org.eclipse.jdt.core.JDTCompilerAdapter"
  classpathref="project.classpath" />
  <javac srcdir="${src}" destdir="${bin}" debug="true"
  source="1.7" target="1.7" encoding="utf-8"
  includeantruntime="false">
    <classpath refid="project.classpath" />
    <ecj />
  </javac>

  <jar destfile="${output}/tests.jar" >
    <fileset dir="${bin}" includes="**/*.class" />
    <fileset dir="${src}" includes="**/*" excludes="**/*
    *.java" />
    <fileset dir="Object Maps" includes="**/*.objectmap"
    erroronmissingdir="false" />
    <fileset dir="Image Assets" includes="**/*.imageasset"
    erroronmissingdir="false" />
    <fileset dir="Verifications" includes="**/*.verification"
    erroronmissingdir="false" />
    <fileset dir="Keyword Assets" includes="**/*.kwd"
    erroronmissingdir="false" />
  </jar>

  <copy todir="${output}" overwrite="true">
    <fileset dir="${lib}" includes="*.jar"
    excludes="*source*" />
  </copy>
  <delete dir="${bin}" />
</target>

<target name="build.keyword.library" depends="compile">
  <java
  classname="com.borland.silk.kwd.library.docbuilder.DocBuilder"
  fork="true">
    <classpath refid="project.classpath" />

    <arg value="AutoQuote Silk4J Library" />
    <arg value="${output}" />
    <arg value="${output}/library.zip" />
  </java>
</target>
</project>

```

通过命令行将关键字库上载到 Silk Central

通过基于 Java 的命令行将外部关键字库上载到 Silk Central，以将 Silk Central 和关键字驱动测试集成到持续集成构建系统，例如 Jenkins。

若要通过基于 Java 的命令行将关键字库上载到 Silk Central：

1. 选择 Silk Central 中的帮助 > 工具，然后下载 **Java 关键字库工具**。
2. 使用以下参数，调用所下载的 jar 文件中包含的命令行工具：

- java
- -jar com.borland.silk.keyworddriven.jar
- -upload
- Silk Central 中要更新的库的库名称，如果尚不存在，则创建一个。
- 要上载的库文件包（zip 存档）的包名称。
- Silk Central 前端服务器的主机名:端口。
- Silk Central 用户的用户名。
- Silk Central 用户的密码。
- 括在引号中的更新信息，描述应用于库的更改。
- [-allowUsedKeywordDeletion]，可选标志，允许删除在测试或关键字序列中使用的关键字。默认情况下，如果试图删除已用的关键字，则会引起错误。

下面的示例概列了将库上载到 Silk Central 的命令行：

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy:Implemented missing keywords"
```

通过自动化手动测试创建关键字驱动测试

在 Silk Central 中将手动测试转换为关键字驱动的测试；


1. 在菜单中，单击**测试 > 详细信息视图**。
2. 在**测试树**中右键单击手动测试，从上下文菜单中选择**自动化方式... > 关键字驱动的测试**。将显示**关键字驱动的测试属性**对话框。
3. 选择包含要使用的关键字的**库**。
必须使用库来存储关键字驱动的测试所使用的一组关键字。
4. 单击**完成**。

将手动测试自动化为关键字驱动的测试将产生以下结果：

- 手动测试转换为自动化关键字驱动的测试，包含针对每个测试步骤的草拟关键字。这些关键字将添加到为该测试选定的库中。
- 如果测试包含对已经为关键字序列存在的共享步骤的调用，则将引用这些关键字序列。
- 如果测试包含对并非为关键字序列的共享步骤的调用，将引用这些共享步骤对象，并将其增强为关键字序列，包含针对每个测试步骤的草拟关键字。这同样适用于嵌套的共享步骤（引用其他共享步骤的共享步骤）。如果嵌套共享步骤的层次结构深度超过 30 级，将为低于该级别的共享步骤的调用创建草拟关键字。
- 如果测试包含对不同库中的共享步骤的调用，则创建并引用这些步骤的草拟关键字。

如果您的测试步骤包含您想在生成的关键字中访问的参数，测试步骤的**操作说明**必须使用正确的语法。参数使用 $\${<name>}$ 语法，例如 $\${username}$ 。在将手动测试转换为关键字驱动的测试时，这些参数将自动添加到所生成的关键字。

在 Silk Central 中创建关键字驱动的测试

1. 在菜单中，单击**测试 > 详细信息视图**。
2. 在要插入新测试的**测试树**中，选择容器或文件夹节点。
3. 在工具栏上单击  (**新建子测试**)，或在树中单击右键并选择**新建子测试**。
新测试节点将追加到树视图，并且将显示**新建测试**对话框。
4. 键入测试的名称和说明。







注: Silk Central 支持 HTML 格式, 并可剪切和粘贴 HTML 内容以用于说明文本框。

5. 从类型列表中选择**关键字驱动**的测试类型。
6. 选择包含要使用的关键字的**库**。
必须使用库来存储关键字驱动的测试所使用的一组关键字。
7. 单击**完成**。

在 Silk Central 内的测试中管理关键字

测试 > 详细信息视图 > <测试> > 关键字

关键字页面使您可以管理选定关键字驱动测试的关键字。可以执行以下操作：

任务	步骤
在 Silk Test 中打开测试或关键字序列	单击用 Silk Test 打开打开 Silk Test 中选定的测试或关键字序列。
添加关键字	<ol style="list-style-type: none"> 1. 单击关键字列表底部的新建关键字, 或者右键单击关键字, 然后从上下文菜单中选择在上面插入关键字。 <p> 注: 可以让 Silk Test 根据关键字用量推荐关键字。用上下文菜单中的启用建议或禁用建议打开或关闭建议。有关更多信息, 请参阅 <i>Silk Test 推荐哪些关键字?</i>。</p> <ol style="list-style-type: none"> 2. 从可用关键字列表选择一个关键字, 或者键入新名称来创建新的关键字。 3. 单击保存。 <p>也可以双击或拖放右侧所有关键字窗格中的现有关键字。</p> <p> 提示: 您可使用 Ctrl+单击 选择多个关键字。在拖动时, 这些关键字将按照您的选择顺序排列。</p>
删除关键字	单击希望删除的关键字的 操作 列中的  。单击 保存 。
更改关键字顺序	将关键字拖放到所需位置。单击 保存 。
创建关键字序列 (一个由其他关键字组成的关键字)	<ol style="list-style-type: none"> 1. 在关键字列表中选择您希望组合的关键字。使用 Ctrl+单击 或 Shift+单击 可选择多个关键字。 2. 右键单击选定内容并单击组合。 3. 为新关键字序列指定名称和说明。
从关键字序列中提取关键字	右键单击一个关键字序列, 单击 提取关键字 。随后, 原始关键字序列将被其中包含的各关键字所取代, 但不会从库中删除。单击 保存 。
为关键字序列定义参数	<ol style="list-style-type: none"> 1. 单击关键字列表上方的参数。此时将显示参数对话框。 2. 单击添加参数。 3. 为新参数指定名称。如果参数为对外参数 (提供一个值, 但不需要输入参数), 请选中输出复选框。 4. 单击确定。 5. 单击保存。
编辑草拟的关键字	<ol style="list-style-type: none"> 1. 单击希望编辑的草拟关键字的操作列中的 。 2. 为关键字选择一个组或者指定一个新组。 3. 键入关键字的说明。这些信息对于将来负责实施关键字的工程师十分有用。 4. 单击确定。

任务	步骤
	<ol style="list-style-type: none"> 5. <i>可选</i>: 在一个参数字段内单击, 为该关键字添加参数。如果关键字使用 Silk Test 实施, 则这些参数将出现在所生成的代码存根中。 6. 单击保存。
搜索关键字	<p>使用右侧的所有关键字窗格查找特定关键字。当您输入字母数字字符时, 列表将使用所有现有匹配项动态更新。搜索技巧:</p> <ul style="list-style-type: none"> • 搜索不区分大小写: doAction 将找到 doaction 和 DOAction。 • 只输入大写字母执行所谓的 <i>驼峰大小写</i> 搜索: ECD 会找到 Enter Car Details、Enter Contact Details 和 EnterContactDetails。 • 搜索时会考虑关键字和组名称: test 将找到包含 test 的所有关键字以及组名称中包含 test 的组中的所有关键字。 • ? 用于取代 0-1 个字符: user?test 将找到 userTest 和 usersTest。 • * 用于取代 0-n 个字符: my*keyword 将找到 myKeyword、myNewKeyword 和 my_other_keyword。 • <字符串>. 仅在组名称中搜索: group. 将找到组名称包含 group 的组中的所有关键字。 • .<字符串> 仅在关键字名称中搜索: .keyword 将找到包含 keyword 的所有关键字。 • <字符串>.<字符串> 搜索特定组中的关键字: group.word 将找到 myGroup 组中的 myKeyword。 • 使用引号搜索完全匹配的项: 'Keyword' 会找到 Keyword 和 MyKeyword, 但不会找到 keyword。

Silk Test 推荐哪些关键字?

在**关键字驱动的测试编辑器**中将关键字添加到关键字驱动的测试或关键字序列时, **Silk Test** 会推荐您可能要用作测试中的下一个关键字的现有关键字。推荐的关键字列在关键字列表的最上面, 并由条形图指出, 条形图填充的部分对应于 **Silk Test** 推荐该关键字的推荐度。

Silk Test 根据以下条件推荐关键字:

- 在第一个关键字添加到关键字驱动的测试或关键字序列时, **Silk Test** 搜索用作其他关键字驱动的测试或关键字序列中的第一个关键字的相似关键字。最常用的关键字推荐度更高。
- 在将更多关键字添加到已经包含其他关键字的关键字驱动的测试或关键字序列时, **Silk Test** 如下推荐关键字:
 - 如果在关键字驱动的测试或关键字序列中, 在您添加新关键字的位置前面已有关键字, 那么 **Silk Test** 将把前面的关键字与所有其他关键字驱动的测试和关键字序列中的关键字组合进行比较, 然后推荐在前面的关键字组合后面最常出现的关键字。
 - 如果在关键字驱动的测试或关键字序列中, 在该位置前面没有关键字, 但是在当前位置后面有关键字, 那么 **Silk Test** 将把后续关键字与所有其他关键字驱动的测试和关键字序列中的关键字组合进行比较, 然后推荐在后续关键字组合前面最常出现的关键字。
- 此外, **Silk Test** 会考虑发现的关键字的相似程度。例如, 如果两个关键字的名称和组都匹配, 那么相比只有名称匹配的两个关键字, **Silk Test** 推荐前两个关键字的可能性更高。
- 如果建立了与 **Silk Central** 的连接, 那么还会考虑包含在关键字驱动的测试中的任何关键字 (属于对应于当前项目的关键字库)。

使用关键字的参数

关键字或关键字序列可以有输入和输出参数。本主题描述如何使用 **Silk Test** 处理这些参数。

在**关键字驱动**的**测试编辑器**中，可以查看某个关键字或关键字序列的任何定义的参数，并且可以编辑参数值。

在**关键字**窗口中，如果将鼠标光标悬停在某个关键字或关键字序列上，就可以看到哪些参数已分配给该关键字或关键字序列。

简单关键字的输入参数

可以定义和使用关键字的输入参数，就像对待任何其他测试方法一样。

下面的代码示例演示了，如何用两个输入参数 `userName` 和 `password` 来定义关键字 `setUserDetails`。

```
@Keyword
public void setUserDetails(String userName, String password) {
    ...
}
```

简单关键字的输出参数

可以定义关键字的返回值或者一个或多个输出参数。还可以使用返回值和一个或多个输出参数的组合。

下面的代码示例演示了如何定义返回字符串的关键字 `getText`：

```
@Keyword
public String getText() {
    return "text";
}
```

下面的代码示例演示了，如何用两个输出参数 `userName` 和 `password` 来定义关键字 `getUserDetails`。

```
@Keyword
public void getUserDetails(OutParameter userName, OutParameter password) {
    userName.setValue("name");
    password.setValue("password");
}
```

关键字序列的参数

可以在**参数**对话框中定义或编辑关键字序列的参数，如果在**关键字序列编辑器**中单击**参数**，就可以打开该对话框。

示例：带参数的关键字

本主题提供了如何使用带参数的关键字的示例。

作为第一步，先创建一个关键字驱动测试，该测试包含您要使用的关键字。可以录制整个关键字驱动测试，或者创建新的关键字驱动测试，然后将关键字添加在关键字驱动测试编辑器中，这样即可创建所要的测试。

在此示例中，关键字驱动测试包含以下关键字：

Start application	这是启动 AUT 并设置基态的标准关键字。
Login	此关键字将特定用户（由用户名和密码标识）登录到 AUT。
GetCurrentUser	此关键字返回当前登录到 AUT 的用户名。
AssertEquals	此关键字比较两个值。

Logout 此关键字将该用户从 AUT 注销。

下一步是将参数添加到关键字。为此，打开关键字的测试脚本，然后将参数添加到方法。

为了将输入参数 `UserName` 和 `Password` 添加到关键字 `Login`，请更改

```
@Keyword("Login")
public void login() {
    ...
}
```

到

```
@Keyword("Login")
public void login(String UserName, String Password) {
    ...
}
```

为了将输出参数 `UserName` 添加到关键字 `GetCurrentUser`，请更改

```
@Keyword("GetCurrentUser")
public void getCurrentUser() {
    ...
}
```






到

```
@Keyword("GetCurrentUser")
public void getCurrentUser(OutParameter CurrentUser) {
    ...
}
```

关键字驱动的测试编辑器中的关键字驱动的测试应类似下面这样：

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<i>CurrentUser ←</i>	
4	 	AssertEquals	<i>Expected</i>	<i>Actual</i>
5	 	Logout		

现在，可以在关键字驱动的测试编辑器中指定输入参数的实际值。若要检索关键字 `GetCurrentUser` 的输出参数 `UserName` 的值，请提供一个变量，例如 `${current user}`。然后可以将存储在变量中的值传给后续关键字。

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<code>\${current user}</code>	
4	 	AssertEquals	John Smith	<code>\${current user}</code>
5	 	Logout		

在 Silk Test 中创建关键字驱动测试

您必须首先选择一个项目，之后才能在 Silk Test 中创建关键字驱动的测试。

使用**关键字驱动的测试编辑器**将新关键字和现有关键字组合为新的关键字驱动的测试。新关键字需要在后续步骤中实现为自动化测试方法。

本主题将介绍如何在 Silk4J 中创建关键字驱动的测试。Silk4NET 和 Silk Test Workbench 的步骤基本类似。有关使用特定 Silk Test 客户端执行关键字驱动的测试的其他信息，请参阅 Silk Test 客户端的文档。

1. 单击 **Silk4J > 新建关键字驱动的测试**。将打开**新建关键字驱动的测试**对话框。
2. 在**名称**字段中为新测试键入名称。
3. 选择应包含新测试的项目。

默认情况下，如果有项目处于活动状态，新测试将在活动项目中创建。



注：为以最优方式利用 Silk Test 提供的功能，除非是在同一个测试中测试多个应用程序，否则就应该为要测试的每个应用程序创建一个独立项目。

4. 单击**完成**。
5. 单击**否**创建空的关键字驱动的测试。将打开**关键字驱动的测试编辑器**。
6. 执行以下操作之一：

- 要添加新关键字，在**新建关键字**字段中键入关键字的名称。
- 要添加现有关键字，展开列表并选择要添加的关键字。

7. 按下 Enter。
8. 重复前两步，直至测试包含要执行的所有关键字为止。
9. 单击**保存**。

继续实施关键字，如果已经实施所有关键字，则继续执行测试。

在 Silk Test 中录制关键字驱动的测试

您必须首先选择一个项目，之后才能在 Silk Test 中创建关键字驱动的测试。

要录制一个关键字，请参阅[录制关键字](#)。

录制新的关键字驱动的测试：


1. 单击 **Silk4J > 新建关键字驱动的测试**。将打开**新建关键字驱动的测试**对话框。
2. 在**名称**字段中为新测试键入名称。
3. 选择应包含新测试的项目。

默认情况下，如果有项目处于活动状态，新测试将在活动项目中创建。



注：为以最优方式利用 Silk Test 提供的功能，除非是在同一个测试中测试多个应用程序，否则就应该为要测试的每个应用程序创建一个独立项目。

4. 单击**完成**。
5. 单击**是**开始录制关键字驱动的测试。
6. 如果您已经为当前项目设置了应用程序配置，并且正在测试 Web 应用程序，将打开**选择浏览器**对话框。选择要在其中录制关键字的浏览器。
7. 根据所打开的对话框，执行以下操作之一：
 - 在**选择应用程序**对话框中，单击**确定**。
 - 在**选择浏览器**对话框中，单击**录制**。

8. 在所测试的应用程序中，执行要包含在第一个关键字内的操作。
有关录制期间可用操作的信息，请参阅 **Silk Test** 客户端的文档。
9. 要为关键字指定名称，将鼠标光标悬停在**录制**窗口内的关键字名称上方，单击**编辑**。
 **注：** **Silk Test** 会自动在关键字驱动的测试的开始位置添加关键字*启动应用程序*。这个关键字将执行应用程序基态，以保证测试成功回放。有关基态的其他信息，请参阅 **Silk Test** 客户端的文档。
10. 在**关键字名称**字段中键入关键字的名称。
11. 单击**确定**。
12. 要为下一个关键字录制操作，在**新关键字名称**字段中键入新关键字的名称，并单击**添加**。**Silk Test** 会将任何新操作录制到新关键字中。
13. 新建关键字并为关键字录制操作，直至为整个关键字驱动的测试录制完成为止。
14. 单击**停止**。此时，**录制完成**对话框将打开。

Silk Test 将使用所有已录制的关键字新建一个关键字驱动的测试。

在 **Silk Test** 中为关键字驱动的测试设置基态。

在使用 **Silk Test** 执行关键字驱动的测试并且关键字驱动的测试调用了基态关键字时，**Silk Test** 将从基态启动您的 AUT。

在录制关键字驱动的测试期间，**Silk Test** 将在当前项目中搜索基态关键字，也就是 `isBaseState` 属性设置为 `true` 的关键字。

- 如果当前项目中存在基态关键字，**Silk Test** 会将此关键字作为关键字驱动的测试中的第一个关键字插入。
- 如果项目中无基态关键字，**Silk Test** 将新建一个名为*启动应用程序*的基态关键字，并将其作为关键字驱动的测试中的第一个关键字插入。

若要手动将关键字标记为基态关键字，请向 **Keyword** 批注添加 `isBaseState` 属性，并将此属性的值设置为 `true`：

```
@Keyword(value = "Start application", isBaseState = true)
public void start_application() {
    // Base state implementation
}
```

在 **Silk Test** 中实施关键字

在实施关键字之前，将关键字定义为某个关键字驱动的测试的一部分。

以下步骤展示了如何在 **Silk4J** 中实施关键字。**Silk4NET** 和 **Silk Test Workbench** 的步骤基本类似。有关使用特定 **Silk Test** 客户端执行关键字驱动的测试的其他信息，请参阅 **Silk Test** 客户端的文档。

实施关键字以便在关键字驱动的测试中重复使用：

1. 打开包含待实施关键字的关键字驱动的测试。
2. 在**关键字驱动的测试编辑器**中，单击要实施的关键字左侧的**实施关键字**。将打开**选择关键字位置**对话框。
3. 单击**选择**以选择要添加关键字实施的程序包和类。
4. **可选：**在**程序包**字段中为新关键字实施定义程序包名称。
5. 在**类**字段中为新关键字实施定义类名称。
6. 单击**确定**。
7. 执行以下操作之一：
 - 要录制关键字，请单击**是**。
 - 要创建空白关键字方法，请单击**否**。

8. 如果您已经为当前项目设置了应用程序配置，并且正在测试 Web 应用程序，将打开**选择浏览器**对话框。选择要在其中录制关键字的浏览器。
9. 单击**录制**。
有关录制的其他信息，请参阅 Silk Test 客户端文档。

如果已实施的关键字在**关键字**窗口中显示为未实施，请在 Eclipse 菜单中选中**项目 > 自动构建**。


在 Silk Test 中实施 Silk Central 关键字

在实施 Silk Central 关键字之前，在 Silk Central 中将关键字定义为关键字驱动的测试的一部分。

以下步骤展示了如何在 Silk4J.Silk4NET 和 Silk Test Workbench 的步骤基本类似。有关使用特定 Silk Test 客户端执行关键字驱动的测试的其他信息，请参阅 Silk Test 客户端的文档。中实施 Silk Central 关键字。

要在 Silk Test 中实施 Silk Central 关键字：

1. 在 Silk Test 中创建与 Silk Central（其中包含关键字驱动的测试）中的关键字库同名的项目。
2. 如果 Silk Central 中的关键字库未指定类型，请单击中的 **Silk4J > 上载关键字库**来设置库类型。
3. *可选*：若要从 Silk Central 在 Silk Test 中实施特定关键字，打开 Silk Central 中库的**关键字**选项卡，然后单击关键字**操作**列中的**用 Silk Test 实施**。
4. 在 Silk Test 菜单中，单击 **Silk4J > 显示关键字视图**。
5. 在**关键字**视图上，双击关键字驱动的测试。
若要来自 Silk Central 的任何更改更新**关键字**视图，请单击**刷新**。
6. 在工具栏中，单击**录制操作**。
7. 如果您已经为当前项目设置了应用程序配置，并且正在测试 Web 应用程序，将打开**选择浏览器**对话框。选择要在其中录制关键字的浏览器。
8. 单击**录制**。
有关录制的其他信息，请参阅 Silk Test 客户端文档。
9. 录制第一个未实施关键字的操作。
10. 在录制了当前关键字的所有操作后，单击**下一个关键字**。
11. 若要在**录制**窗口中切换关键字，请单击**上一个关键字**和**下一个关键字**。
12. 单击**停止**。此时，**录制完成**对话框将打开。

 **注**：不能从 Silk Central 删除关键字驱动的测试中的关键字，也不能更改其关键字顺序，因为这些测试在 Silk Test 中只读。

如果已实施的关键字在**关键字**窗口中显示为未实施，请在 Eclipse 菜单中选中**项目 > 自动构建**。

在 Silk Test 中录制关键字

仅能为关键字驱动的测试中已经存在的关键字录制操作，而不能为全新的关键字录制。要录制新关键字驱动的测试，请参阅[录制关键字驱动的测试](#)。


为新关键字录制操作：

1. 打开包含待录制关键字的关键字驱动的测试。
2. 在**关键字驱动的测试编辑器**中，单击要实施的关键字左侧的**实施关键字**。将打开**选择关键字位置**对话框。
3. 单击**选择**以选择要添加关键字实施的程序包和类。
4. *可选*：在**程序包**字段中为新关键字实施定义程序包名称。
5. 在**类**字段中为新关键字实施定义类名称。
6. 单击**确定**。

7. 如果您已经为当前项目设置了应用程序配置，并且正在测试 Web 应用程序，将打开**选择浏览器**对话框。选择要在其中录制关键字的浏览器。
8. 单击**录制**。将打开**录制**对话框，**Silk Test** 开始为该关键字录制操作。
9. 在所测试的应用程序中，执行要测试的操作。
有关录制期间可用操作的信息，请参阅 **Silk Test** 客户端的文档。
10. 单击**停止**。**录制完成**对话框随即打开。

已录制的操作将在指定类的上下文中显示。

编辑关键字驱动的测试

 **注：**在 **Silk Test** 中，可以编辑和执行位于 **Silk Test** 中的关键字驱动的测试，并且可以执行存储在 **Silk Central** 中的关键字驱动的测试。若要编辑存储在 **Silk Central** 中的关键字驱动的测试，请在**关键字驱动的测试编辑器**中打开关键字驱动的测试，然后单击**编辑**。

编辑关键字驱动的测试：

1. 在**关键字驱动的测试编辑器**中打开关键字驱动的测试。
 - a) 在 **Silk Central** 菜单的**测试 > 详细信息视图**中，展开关键字驱动的测试所在的项目。
 - b) 在**测试树**中选择关键字驱动的测试。
 - c) 选择**关键字**选项卡。

2. 为关键字驱动的测试添加新关键字：

- a) 在**新建关键字**字段内单击。
- b) 为新关键字键入名称。
- c) 按下 **Enter**。

3. 要编辑现有关键字，单击关键字左侧的**打开关键字**。

 **注：****Silk Central** 具有在 **Silk Central** 中创建的任何关键字的所有权，也就是说您对此类关键字所做的任何更改都将保存在 **Silk Central** 中，而非 **Silk Test** 中。


4. 要从关键字驱动的测试中删除关键字，请单击关键字左侧的**删除关键字**。
关键字仍然在**关键字**窗口中可用，您可以随时重新将其添加到关键字驱动的测试。
5. 若要保存更改，请单击**保存**。

将关键字组合为关键字序列

对于多个关键字驱动的测试中希望顺序执行的关键字，可以使用**关键字驱动的测试编辑器**将其组合为一个关键字序列。


1. 打开包含待组合关键字的关键字驱动的测试。
2. 在**关键字驱动的测试编辑器**中，按住 **Ctrl** 键，然后单击要组合的关键字。
3. 右键单击选定内容，单击**组合**。此时将打开**组合关键字**对话框。
4. 在**名称**字段中键入新关键字序列的名称。
5. **可选：**在**说明**字段中键入新关键字序列的说明。
6. 单击**组合**。

新关键字序列将打开，同时也会显示在**关键字**窗口中。您可以在关键字驱动的测试中使用关键字序列。

 **注：**与其他任何关键字相同，关键字序列不能单独执行，只能作为关键字驱动的测试的一部分执行。

从 Eclipse 回放关键字驱动的测试

以下步骤展示了如何在 Silk4J 中回放关键字驱动的测试。Silk4NET 和 Silk Test Workbench 的步骤基本类似。有关使用特定 Silk Test 客户端执行关键字驱动的测试的其他信息，请参阅 Silk Test 客户端的文档。

1. 在**项目资源管理器**中，导航到要回放的关键字驱动的测试资产。
2. 右键单击资产名称。
3. 选择**运行方式 > 关键字驱动的测试**。
4. *可选*：若要打开**运行配置**对话框，请选择**运行方式 > 运行配置**。
5. *可选*：在**运行配置**对话框中，可以选择不同的测试或项目。
6. *可选*：在**运行配置**对话框的**全局变量**网格中，可以设置用于执行关键字驱动的测试的任何变量的值。每当您执行关键字驱动的测试资产时，都会用到这些值。
 - a) 将变量的**变量名称**和**值**键入相应字段。
 - b) 键入 **Enter** 将新行添加到网格。
 - c) 重复前面两步，直至设置完您要使用的全部全局变量的值。在执行属于自动化框架一部分，且在测试管理工具（例如 Silk Central）中管理的关键字驱动的测试时，还可以将新的 `.properties` 文件添加到项目，以设置整个项目的全局变量的值。有关其他信息，请参见**使用特定变量回放关键字驱动的测试**。
7. *可选*：若要关闭**运行配置**对话框，并开始执行关键字驱动的测试资产，请单击**运行**。
8. 如果您正在测试 Web 应用程序，将打开**选择浏览器**对话框。选择浏览器，然后单击**运行**。
 **注**：如果为当前配置了多个应用程序，则不显示**选择浏览器**对话框。
9. *可选*：如有必要，可以同时按下两个 **Shift** 键，停止执行测试。
10. 测试执行完成时，将打开**回放完成**对话框。单击**浏览结果**，查看已完成测试的 TrueLog。

从 Silk Central 回放 Silk Test 测试

要从 Silk Central 访问 Silk Test 测试，您需要将 Silk Test 测试存储在 Silk Central 可通过源代码控制配置文件访问的存储库内的 JAR 文件中。

要从 Silk Central 回放 Silk Test 中的功能测试，例如关键字驱动的测试：

1. 在 Silk Central 中，创建将从中执行 Silk Test 测试的项目。
2. 在**测试 > 详细信息视图**下，为新项目新建一个测试容器。
有关 Silk Central 的其他信息，请参阅 [Silk Central 帮助](#)。
要为 Silk Test 测试指定源代码控制配置文件，必须使用测试容器。
 - a) 在**测试树**中，右键单击要在其下添加新测试容器的节点。
 - b) 单击**新建测试容器**。将打开**新建测试容器**对话框。
 - c) 在**名称**字段中键入新测试容器的名称。
例如，键入关键字驱动的测试
 - d) 在**源代码控制配置文件**字段中，选择包含 Silk Test 测试的 JAR 文件所在的源代码控制配置文件。
 - e) 单击**确定**。
3. 在新测试容器中新建一个 JUnit 测试。
有关 Silk Central 的其他信息，请参阅 [Silk Central 帮助](#)。
 - a) 在**JUnit 测试属性**对话框的**测试类**字段中，键入测试类的名称。
指定测试套件类的完全限定名称。有关其他信息，请参见[通过命令行回放关键字驱动的测试](#)。
 - b) 在**类路径**字段中，指定包含测试的 JAR 文件的名称。

c) 对于关键字驱动测试，还应指定以下文件的路径（用分号分隔）。

- com.borland.silk.keyworddriven.engine.jar
- com.borland.silk.keyworddriven.jar
- silktest-jtf-nodeps.jar

这些文件位于 Silk Test 安装目录中。例如，JAR 文件 tests.jar 中的关键字驱动的测试的类路径字段可能如下所示：

```
tests.jar;C:\Program Files
(x86)\Silk\SilkTest\ng\KeywordDrivenTesting
\com.borland.silk.keyworddriven.engine.jar;C:\Program Files
(x86)\Silk\SilkTest\ng\KeywordDrivenTesting
\com.borland.silk.keyworddriven.jar;C:\Program Files
(x86)\Silk\SilkTest\ng\JTF\silktest-jtf-nodeps.jar
```

4. 单击完成。

5. 执行测试。

有关在 Silk Central 中执行测试的其他信息，请参阅 [Silk Central 帮助](#)。

通过命令行回放关键字驱动的测试

在执行此任务之前，您必须更新 PATH 变量，以引用您的 JDK 位置。有关详细信息，请参阅如下 Sun 文档：<http://java.sun.com/j2se/1.5.0/install-windows.html>。

要从命令行回放关键字驱动的测试，例如从 CI 服务器回放测试时，请使用 KeywordTestSuite 类。

1. 为了从命令行执行关键字驱动的测试，请创建一个带有 @KeywordTests 批注的 JUnit 测试套件。例如，如果要执行关键字驱动的测试 *My Keyword-Driven Test*，请创建如下的 JUnit 测试套件 MyTestSuite：

```
@RunWith(KeywordTestSuite.class)
@KeywordTests({ "My Keyword-Driven Test" })
public class MyTestSuite {
}
}
```

2. 在 CLASSPATH 中包含以下内容：

- junit.jar。
- org.hamcrest.core JAR 文件。
- silktest-jtf-nodeps.jar。
- com.borland.silk.keyworddriven.engine.jar。
- 包含关键字驱动的测试的文件夹的 JAR。

```
set CLASSPATH=<eclipse_install_directory>\plugins
\org.junit_4.11.0.v201303080030\junit.jar;<eclipse_install_directory>
\plugins\org.hamcrest.core_1.3.0.v201303031735.jar;%OPEN_AGENT_HOME%\JTF
\silktest-jtf-nodeps.jar;%OPEN_AGENT_HOME%\KeywordDrivenTesting
\com.borland.silk.keyworddriven.engine.jar;C:\myTests.jar
```

3. 可选：将新的 .properties 文件添加到项目，以设置用于执行该关键字驱动的测试的任何变量的值。

有关其他信息，请参见 [使用特定变量回放关键字驱动的测试](#)。

4. 键入 java org.junit.runner.JUnitCore <Name> 来运行 JUnit 测试方法，其中 Name 是在第一步中创建的 JUnit 测试套件的名称。



注：有关故障排除信息，请参阅以下 JUnit 文档：http://junit.sourceforge.net/doc/faq/faq.htm#running_1。

示例

例如, 要运行 *My Keyword Driven Test 1* 和 *My Keyword Driven Test 2* 这两个关键字驱动测试, 请创建以下类:

```
package demo;

import org.junit.runner.RunWith;

import com.borland.silktest.jtf.keyworddriven.KeywordTestSuite;
import com.borland.silktest.jtf.keyworddriven.KeywordTests;

@RunWith(KeywordTestSuite.class)
@KeywordTests({ "My Keyword Driven Test 1", "My Keyword Driven
Test 2" })
public class MyTestSuite {

}
```

若要从命令行运行该类, 请键入如下代码:

```
java org.junit.runner.JUnitCore demo.KeywordTestSuite
```

若要使用存储在 `c:\temp\globalvariables.properties` 文件中的全局变量, 从命令行运行该类, 请键入以下代码:

```
java -Dsilk.keyworddriven.engine.globalVariablesFile=c:\temp
\globalvariables.properties org.junit.runner.JUnitCore
demo.KeywordTestSuite
```

有关其他信息, 请参见 [使用特定变量回放关键字驱动测试](#)。

用 Apache Ant 回放关键字驱动测试

为了执行本主题中所述的操作, 请确保计算机上安装了 Apache Ant。

若要用 Apache Ant 回放关键字驱动测试, 以生成测试运行情况的 HTML 报告, 请使用 `KeywordTestSuite` 类。

1. 为了用 Apache Ant 执行关键字驱动测试, 请创建一个带有 `@KeywordTests` 批注的 JUnit 测试套件。例如, 如果要执行关键字驱动测试 *My Keyword-Driven Test*, 请创建如下的 JUnit 测试套件 `MyTestSuite`:

```
@RunWith(KeywordTestSuite.class)
@KeywordTests({ "My Keyword-Driven Test" })
public class MyTestSuite {

}
```

2. 打开 Silk Test 项目的 `build.xml` 文件, 该项目包含关键字驱动测试。
3. 为了执行关键字驱动测试, 请将以下目标添加到 `build.xml` 文件:

```
<target name="runTests" depends="compile">
  <mkdir dir="./reports"/>
  <junit printsummary="true" showoutput="true" fork="true">
    <classpath>
      <fileset dir="${output}">
        <include name="**/*.jar" />
      </fileset>
      <fileset dir="${buildlib}">
        <include name="**/*.jar" />
      </fileset>
      <fileset dir="C:/Program Files (x86)/Silk/SilkTest/ng/
KeywordDrivenTesting">
```

```

    <include name="**/*.jar" />
  </fileset>
</classpath>

  <test name="MyTestSuite" todir="./reports"/>
</junit>
</target>

```

有关 JUnit 任务的更多信息，请参阅 <https://ant.apache.org/manual/Tasks/junit.html>。

4. 可选：若要创建所有测试的 XML 报告，请将以下代码添加到目标：

```
<formatter type="xml" />
```

5. 可选：若要根据 XML 报告创建 HTML 报告，请将以下代码添加到目标：

```

<junitreport todir="./reports">
  <fileset dir="./reports">
    <include name="TEST-*.xml" />
  </fileset>
  <report format="noframes" todir="./report/html" />
</junitreport>

```

有关 JUnitReport 任务的更多信息，请参阅 <https://ant.apache.org/manual/Tasks/junitreport.html>。

整个目标现在应该类似于下面：

```

<target name="runTests" depends="compile">
  <mkdir dir="./reports"/>
  <junit printsummary="true" showoutput="true" fork="true">
    <classpath>
      <fileset dir="${output}">
        <include name="**/*.jar" />
      </fileset>
      <fileset dir="${buildlib}">
        <include name="**/*.jar" />
      </fileset>
      <fileset dir="C:/Program Files (x86)/Silk/SilkTest/ng/
KeywordDrivenTesting">
        <include name="**/*.jar" />
      </fileset>
    </classpath>

    <formatter type="xml" />

    <test name="MyTestSuite" todir="./reports"/>
  </junit>
  <junitreport todir="./reports">
    <fileset dir="./reports">
      <include name="TEST-*.xml" />
    </fileset>
    <report format="noframes" todir="./report/html" />
  </junitreport>
</target>

```

6. 若要从 Eclipse 运行测试，请执行以下操作：
- 在**程序包管理器**中，右键单击 build.xml 文件。
 - 选择**运行方式 > Ant 构建...**
 - 在**编辑配置**对话框的**目标**选项卡中，选中 **runTests**。
 - 单击**运行 (Run)**。

还可以从命令行或 CI 服务器执行测试。有关更多信息，请参阅 *Silk Test 帮助*中的 <https://ant.apache.org/manual/running.html> 以及从**持续集成服务器回放测试**。

使用特定变量回放关键字驱动的测试

您必须首先创建项目，然后才能为执行关键字驱动的测试而设置变量值。

若要为关键字驱动的测试资产的所有执行设置全局变量的值（这些执行都由您触发），请使用**运行配置**对话框中的**全局变量**网格。有关其他信息，请参见从 *Eclipse* [回放关键字驱动的测试](#)。

在执行属于自动化框架一部分、在测试管理工具（例如 **Silk Central**）中管理的关键字驱动的测试时，您可以在 **Silk Test** 中设置用于执行关键字驱动的测试的任何变量的值。若要设置整个项目的全局变量的值（这意味着，每当 **Silk Test** 用户执行此项目中关键字驱动的测试时，都会用到这些值），请执行以下操作：

1. 在**程序包管理器**中，展开包含要根据变量执行的关键字驱动的测试的项目。
2. 右键单击项目的**关键字驱动的测试**文件夹，选择**新建 > 文件**。此时将打开**新建文件**对话框。
3. 在**文件名称**字段中键入 `globalvariables.properties`。
4. 单击**完成**。将打开新的属性文件。
5. 在文件中添加新行，以指定变量。

新变量的格式如下：

```
name=value
```

例如，要指定 *user* 和 *password* 这两个变量，请键入以下内容：

```
user=John  
password=john5673
```

有关属性文件格式以及如何输入 **UNICODE** 字符（例如空格）的信息，请参阅[属性文件格式](#)。

6. 保存 `globalvariables.properties` 文件。

只要项目中有关关键字驱动的测试，就要使用变量。

分组关键字

为了更好地设置关键字在库中的结构，您可以对其进行分组。

本主题介绍如何将关键字添加到 **Silk4J** 中的指定组内。**Silk4NET** 和 **Silk Test Workbench** 的步骤基本类似。有关使用特定 **Silk Test** 客户端执行关键字驱动的测试的其他信息，请参阅 **Silk Test** 客户端的文档。这些组名称也供 **Silk Central** 使用，您的关键字将相应地分组。

向特定组添加关键字：

1. 打开关键字的实施。
 - a) 打开实施了关键字的项目。
 - b) 打开**关键字**窗口。
 - c) 在**关键字**窗口中，选择关键字。
 - d) 单击**转到实施**。
2. 要将一个类中的所有方法添加到关键字组中，请将关键字组添加到类开始位置之前。

例如，要为关键字添加组计算器，请键入：

```
@KeywordGroup("Calculator")
```

在**关键字**窗口中，所显示的关键字现将包含分组。例如，*Calculator* 组中的关键字 *Addition* 将显示为 `Calculator.Addition`。

索引

A

Ant

运行关键字驱动测试 20

B

变量

执行关键字驱动测试 22

C

参数

处理, 关键字 12

测试自动化

概述 4

持续集成

正在上载关键字库 8

创建

关键字驱动测试 9, 14

D

打开

关键字 10

F

分组

关键字 22

G

概念

测试自动化 4

关键字

参数 10, 12

参数, 示例 12

打开 10

分组 22

关于 5

管理 10

录制 16

嵌套 10

删除 10

上载到 Silk Central 6

实施 10, 15

替换 10

添加 10

序列 10

组合 10, 17

关键字库

上载 8

关键字驱动的

测试 4

关键字驱动测试

编辑 17

参数, 示例 12

创建 9, 14

从 Silk Central 执行 18

从命令行运行 19

概述 4

关键字推荐, 算法 11

回放 18

基态 15

录制 14

删除关键字 17

实施 Silk Central 关键字 16

实施关键字 15

添加关键字 17

应用程序配置 15

用 Ant 运行 20

优势 4

正在上载关键字, Silk Central 6

正在停止 18

指定变量, 执行 22

转换自手动测试 9

关键字驱动的测试编辑器

推荐的关键字 11

关键字序列

参数 12

正在创建 17

管理

关键字 10

J

基态

关键字驱动测试 15

集成

配置 Silk Central 位置 6

K

库

上载 8

L

录制

关键字 16

关键字驱动测试 14

M

命令行

运行关键字驱动测试 19

Q

嵌套

关键字 10

S

Silk Central

- 配置位置 6
- 运行测试 18
- 正在上载关键字 6

Silk Central 关键字

实施 16

删除

关键字 10

删除关键字

关键字驱动测试 17

上载

关键字库 8
库 8

实施

关键字 10

手动测试

转换为关键字驱动测试 9

T

替换

关键字 10

添加

关键字 10

添加关键字

关键字驱动测试 17

推荐

算法 11

推荐的关键字

关键字驱动测试编辑器 11

Y

应用程序配置

关键字驱动测试 15

运行测试

Silk Central 18

Z

正在停止

运行关键字驱动测试 18

执行关键字驱动测试

变量 22

转换

手动测试到关键字驱动测试 9

组合

关键字 10