



Silk Test 18.0

キーワード駆動テスト

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

Copyright © Micro Focus 1992-2017. All rights reserved.

MICRO FOCUS, Micro Focus ロゴ及び Silk Test は Micro Focus IP Development Limited
またはその米国、英国、その他の国に存在する子会社・関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2017-06-12

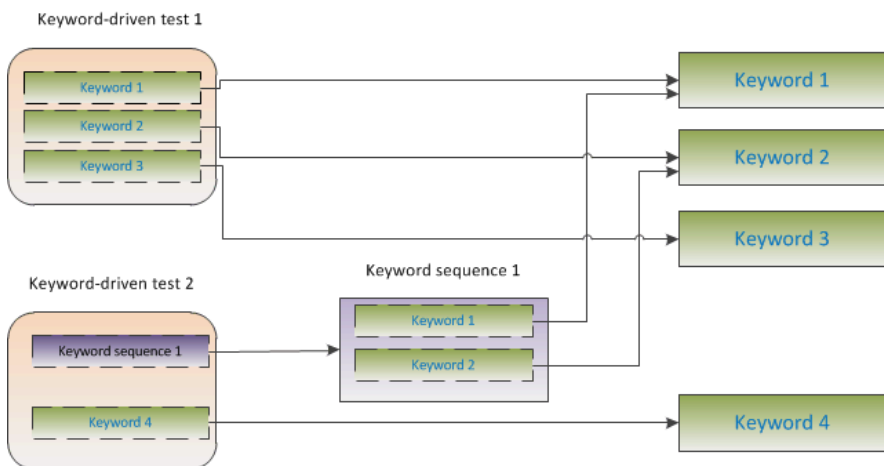
目次

キーワード駆動テスト	4
テスト自動化の概要	4
キーワード駆動テストの利点	5
キーワード	5
Silk Test と Silk Central の統合	6
Silk Central へのキーワード ライブラリのアップロード	7
コマンド ラインから Silk Central へのキーワード ライブラリの更新	9
手動テストを自動化してキーワード駆動テストを作成する	10
Silk Central でキーワード駆動テストを作成する	10
Silk Central でテストのキーワードを管理する	11
Silk Test のキーワード レコメンド機能	13
キーワードでのパラメータの使用	13
例：パラメータを取るキーワード	14
Silk Test でキーワード駆動テストを作成する	16
Silk Test でのキーワード駆動テストの記録	16
Silk Test でのキーワード駆動テストの基本状態の設定	17
Silk Test でのキーワードの実装	18
Silk Test での Silk Central キーワードの実装	18
Silk Test でのキーワードの記録	19
キーワード駆動テストの編集	20
キーワードのキーワード シーケンスへの結合	21
Eclipse からのキーワード駆動テストの再生	21
Silk Central からの Silk Test テストの再生	22
コマンド ラインからのキーワード駆動テストの再生	23
Apache Ant を使用したキーワード駆動テストの再生	24
変数を指定したキーワード駆動テストの再生	25
キーワードのグループ化	26

キーワード駆動テスト


キーワード駆動テストは、テスト開発からテスト設計を分離するソフトウェアテスト手法です。このため、テスト自動化プロセスにビジネスアナリストなどの専門家を含めることができます。Silk Central と Silk Test はキーワード駆動テストをサポートしており、Silk Test のキーワードの形式での共有資産として構成されるメンテナンス可能な自動化フレームワークを自動化エンジニアが開発することによって、自動化エンジニアとビジネスアナリスト間で密接な共同作業を行うことができます。その後、これらのキーワードは、Silk Test で新しいキーワード駆動テストを作成したり、Silk Central で既存の手動テスト資産を自動テストに変換したり、新しいキーワード駆動テストを作成するために、ビジネスアナリストが使用することができます。

- キーワード駆動テストは、実行可能なキーワードのコレクションです。キーワード駆動テストは、他のテストと同様に再生することができます。
- キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを1つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。
- キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。



キーワード駆動テストの作成には、次の2つのフェーズがあります。

1. テストの設計
2. キーワードの実装

 **注:** 以下のトピックでは、Silk Central を使用して手動テストを自動化する方法、および Silk Central や Silk Test を使用してキーワード駆動テストを実行する方法について説明します。Silk Test で実行されるタスクは、Eclipse の Silk Test プラグインである Silk4J をベースに説明します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。

テスト自動化の概要

テスト自動化とは、実行の実際の結果と期待する結果を比較する検証を使用して、テストの実行を制御するためにソフトウェアを使用する処理です。Silk Test のコンテキストでは、テスト自動化には機能テストと回帰テストの自動化を含めます。機能テストと回帰テストを自動化すると、コストと時間、およびリソースを節約できます。

テスト自動化フレームワークとは、特定の製品の自動化の規則を集めた統合システムです。このシステムは、テスト、テストデータソース、オブジェクトの詳細、およびさまざまな再利用可能モジュールを統合します。これらのコンポーネントは、ビジネスプロセスを表現するために組み合わせて使用されます。フレームワークによって、テスト自動化の基礎が提供され、自動化の労力も低減されます。

キーワード駆動テストの利点

キーワード駆動テスト手法を使用する利点を次に示します。

- キーワード駆動テストを使用すると、テスト自動化とテストケースのデザインが分離され、うまく分業できるようになり、キーワードを実装するテストエンジニアとテストケースをデザインする専門家が共同作業できます。
- テスト対象アプリケーションにアクセスすることなく、初期の段階からテストを開発でき、後からキーワードを実装できます。
- プログラムの知識がなくてもテストを開発できます。
- キーワード駆動テストは、長期的に見るとメンテナンスコストを低減できます。キーワードのメンテナンスが必要で、これらのキーワードを使用するすべてのキーワード駆動テストは自動的に更新されます。
- テストケースが簡潔です。
- 技術者でなくてもテストケースが読みやすく、理解しやすくなります。
- テストケースの変更が簡単です。
- 既存のキーワードを再利用して新しいテストを再利用できます。これにより、より広範囲なテストカバレッジを実現しやすくなります。
- キーワード実装の内部的な複雑性を、キーワード駆動テストを作成または実行するユーザーに対して隠蔽できます。

キーワード

キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。Silk Test でのキーワードは、アノテーション付きのテストメソッド (@Keyword) です。キーワードは、キーワード資産として保存されます。

キーワード駆動テストの作成中にキーワードやキーワードシーケンスを定義し、後でそれらをテストメソッドとして実装できます。既存のテストメソッドに @Keyword アノテーションを付けて、キーワードとしてマークすることもできます。Java では、キーワードは次のアノテーションで定義されます：


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

キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを 1 つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。

キーワードは入力パラメータと出力パラメータを持つことができます。キーワードを実装するテストメソッドのパラメータは、キーワードのパラメータです。キーワードのパラメータに違う名前を指定するために、次を使用できます：

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

デフォルトでは、パラメータは Silk4J の入力パラメータです。出力パラメータを定義するには、OutParameter クラスを使用します。

 **注：キーワード駆動テストエディタ** でキーワードの出力パラメータを指定するには、次のように記述します。

```
${parameter_name}
```

キーワード駆動テスト エディタ で、キーワードの出力パラメータを他のキーワードの入力パラメータとして使用する場合も、同じように記述します。

例

キーワードとしてマークされたテストメソッドは、次のようになります。

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

または

```
// 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
}
```

このキーワードは、指定したユーザー名とパスワードを使ってテスト対象アプリケーションにログインし、ログインが成功したかどうかを返します。出力パラメータを他のキーワードの入力パラメータとして使用するには、キーワード内で出力パラメータに値を設定します。



注: ヘルプ トピックを PDF で参照している場合、このサンプルコードは、実際のスクリプトでは許されない場所で改行されてしまっている場合があります。スクリプトでこのサンプルコードを使用する場合は、これらの改行を削除してください。

- **Keyword** アノテーションのキーワード名パラメータは、省略可能です。メソッドの名前とは異なる名前を指定する場合に、キーワード名パラメータを使用できます。パラメータが指定されていない場合、メソッドの名前がキーワード名として使用されます。
- **Argument** アノテーションも省略可能です。メソッドをキーワードとしてマークすると、自動的にすべての引数がキーワードの引数として使用されます。例のように、*userName* を *UserName* にしたい場合など、キーワードの引数とは異なる名前を指定する場合に、**Argument** アノテーションを使用できます。

Silk Test と Silk Central の統合

Silk Test と Silk Central を統合することによって、技術者と非技術者のユーザー間で共同作業を行えます。

Silk Test と Silk Central が統合され、Silk Central に存在するライブラリとアクティブな Silk Test プロジェクトが同じ名前であれば、**キーワード ビュー** (**Silk4J** > **キーワード ビューの表示** から開く) には、アクティブな Silk Test プロジェクトで定義されたキーワードに加えて、Silk Central ライブラリのすべてのキーワードが表示されます。




注: Silk Central の接続情報は、すべての Silk Test ユーザーに別々に保存されるため、Silk Central のキーワードおよびキーワードシーケンスで作業するすべての Silk Test ユーザーは、Silk Test を Silk Central と統合する必要があります。

Silk Test と Silk Central を統合すると、次のメリットがあります。

- テスト管理と実行を Silk Central で処理できる

- キーワードが Silk Central データベースに格納され (ライブラリのアップロード)、Silk Central のすべてのプロジェクトで利用できる
- 手動テストを Silk Central で直接自動化し、作成したキーワード駆動テストを Silk Central から Silk Test で実行できる

 **注:** Silk Test では、Silk Test にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テスト エディター** でキーワード駆動テストを開き、**編集** をクリックします。

次の手順は、Silk4J と Silk Central を統合する方法を示します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。


1. メニューから、**Silk4J > Silk Central の設定** を選択します。**設定** ダイアログ ボックスが開きます。
2. **URL** フィールドに、Silk Central サーバーの URL を入力します。
たとえば、Silk Central のサーバー名が *sctm-server* で、Silk Central のポート番号が 13450 の場合は、`http://sctm-server:13450` と入力します。
3. 有効な ユーザー名 と パスワード を、それぞれのフィールドに入力します。
4. **検証** をクリックして、Silk Test が指定したユーザーで Silk Central サーバーにアクセスできるかどうか確認します。
5. **OK** をクリックします。

Silk Central へのキーワード ライブラリのアップロード

Silk Central で作業するためには、有効な Silk Central の場所が設定されている必要があります。詳細については、「[Silk Test と Silk Central の統合](#)」を参照してください。

Silk Central で自動テストを自動化するために、Silk Test プロジェクトで実装したキーワードをキーワード ライブラリとして Silk Central にアップロードできます。そして、キーワードを使用して手動テストを自動化できます。

1. Silk Test で、キーワード駆動テストが存在するプロジェクトを選択します。
2. 同じ名前のライブラリが Silk Central (**テスト > ライブラリ**) に存在していることを確認します。
3. ツールバーで、**キーワード ライブラリのアップロード** をクリックします。
4. 省略可能：キーワード ライブラリに対する変更の説明を指定します。
5. 省略可能：**設定** をクリックして、Silk Central への接続を設定します。
6. 省略可能：接続した Silk Central インスタンスで利用可能なライブラリを確認するには、リンクをクリックします。
7. **アップロード** をクリックします。

 **注意:** Silk Central のキーワード ライブラリが既に他の自動化ツール、つまり他の Silk Test クライアントに割り当てられている場合、キーワード ライブラリのタイプを変更するかどうか確認されます。タイプの変更を行う場合にのみ、ライブラリはアップロードされます。

Silk Test は、プロジェクトで実装されたすべてのキーワードからキーワード ライブラリを作成します。その後、Silk Test は `library.zip` という名前で、キーワード ライブラリをプロジェクトの出力フォルダに保存します。ライブラリの整合性が検証され、Silk Central の既存のテストが影響を受ける変更が **キーワード ライブラリを Silk Central にアップロード** ダイアログ ボックスに一覧されます。最後に、Silk Test はライブラリを Silk Central にアップロードします。これで、Silk Central でキーワードを使用できるようになります。キーワード ライブラリに含まれるキーワードを使用する Silk Central のキーワード駆動テストは、現在のキーワードの実装を自動的に使用します。

Silk Test 15.5 で作成したプロジェクトからのキーワード ライブラリのアップロード

Silk Test 15.5 で作成した Silk Test プロジェクトからキーワード ライブラ리를アップロードする場合は、プロジェクトの build.xml ファイルを編集する必要があります。

1. **パッケージ・エクスプローラー** で、キーワード ライブラ리를アップロードするプロジェクトのフォルダを展開します。
2. build.xml ファイルを開きます。
3. プロジェクトの Keyword Assets ディレクトリを *compile* ターゲットの JAR ビルドステップに追加します。

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

4. キーワード ライブラリ用の次のターゲットを追加します。

```
<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>
```

新しい build.xml ファイルは、以下のようになります。

```
<?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="buildlib" value="buildlib" />

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

```

コマンド ラインから Silk Central へのキーワード ライブラリの更新

Java ベースのコマンド ラインから外部キーワード ライブラリを Silk Central にアップロードすることで、Silk Central とキーワード駆動テストを Jenkins などの継続的インテグレーション ビルド システムに統合することができます。

Java ベースのコマンド ラインからキーワード ライブラリを Silk Central にアップロードするには :

1. Silk Central で、ヘルプ > ツール を選択し、Java キーワード ライブラリ ツール をダウンロードします。
2. ダウンロードした jar ファイルに含まれるコマンド ライン ツールを次の引数で呼び出します。
 - java
 - -jar com.borland.silk.keyworddriven.jar
 - -upload
 - 更新または存在しない場合は作成される Silk Central のライブラリの ライブラリ名。
 - アップロードするライブラリ パッケージ (zip アーカイブ) のパッケージ名。
 - Silk Central フロントエンド サーバーのホスト名:ポート。
 - Silk Central ユーザーの ユーザー名。
 - Silk Central ユーザーのパスワード。
 - ライブラリに適用された変更を説明する 更新情報 (引用符で囲む)。

- テストまたはキーワード シーケンスで使用されているキーワードの削除を許可するための [-allowUsedKeywordDeletion] フラグ (省略可能)。デフォルトでは、使用中のキーワードを削除しようとするとエラーが発生します。

ライブラリを Silk Central にアップロードするコマンド ラインは次のようになります。

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

手動テストを自動化してキーワード駆動テストを作成する

Silk Central の手動テストをキーワード駆動テストに変換するには :


1. メニューで **テスト > 詳細ビュー** をクリックします。
2. **テスト ツリー** で手動テストを右クリックし、コンテキスト メニューから **自動化... > キーワード駆動テスト** を選択します。 **キーワード駆動テスト プロパティ** ダイアログ ボックスが開きます。
3. 使用したいキーワードを含む **ライブラリ** を選択します。
ライブラリは、キーワード駆動テストが使用するキーワード群を格納するために必要です。
4. **終了** をクリックします。


手動テストを自動化する、次のようなキーワード駆動テストが生成されます。

- 手動テストは、各テスト ステップに対してドラフト キーワードを含んだ自動化されたキーワード駆動テストに変換されます。これらのキーワードは、このテストに対して選択されたライブラリに追加されます。
- テストに、すでにキーワード シーケンスとして存在する共有ステップへの呼び出しが含まれている場合、そのキーワード シーケンスが参照されます。
- テストに、キーワード シーケンスとして存在しない共有ステップへの呼び出しが含まれている場合、その共有ステップ オブジェクトは、各テスト ステップに対してドラフト キーワードを含んだキーワード シーケンスに変換され、参照されます。これは、ネストした共有ステップ (他の共有ステップを参照する共有ステップ) にも適用されます。ネストした共有ステップの階層が 30 レベルを超えた深さの場合、そのレベル以下の共有ステップへの呼び出しに対してドラフト キーワードが作成されます。
- テストが他のライブラリにある共有ステップへの呼び出しを含んでいる場合、そのステップに対するドラフト キーワードが作成され、参照されます。

テスト ステップに、生成したキーワードでアクセスしたいパラメータが含まれている場合、テスト ステップの **アクションの説明** で、正しい文法を使用する必要があります。パラメータは、`${<名前>}` という構文を使用します (`${username}` など)。手動テストをキーワード駆動テストに変換する際には、これらのパラメータは生成されたキーワードに自動的に追加されます。

Silk Central でキーワード駆動テストを作成する

1. メニューで **テスト > 詳細ビュー** をクリックします。
2. 新規のテストの挿入先となるコンテナまたはフォルダを、**テスト ツリー** から選択します。
3. ツールバーの  (**子テストの新規作成**) をクリックするか、ツリー内を右クリックし、**子テストの新規作成** を選択します。
新規のテスト ノードがツリー ビューに追加され、**テストの新規作成** ダイアログ ボックスが表示されます。
4. [名前] フィールドと [説明] フィールドに、テストの名前と説明をそれぞれ入力します。




 **注:** Silk Central では、HTML 書式をサポートしており、**説明** フィールドへの HTML コンテンツの切り取り/貼り付けが可能です。



5. **タイプ** リストから **キーワード駆動テスト** テスト タイプを選択します。
6. 使用したいキーワードを含む **ライブラリ** を選択します。
ライブラリは、キーワード駆動テストが使用するキーワード群を格納するために必要です。
7. **完了** をクリックします。

Silk Central でテストのキーワードを管理する

テスト > 詳細ビュー > <テスト> > キーワード

キーワード ページでは、選択したキーワード駆動テストのキーワードを管理できます。次のアクションを実行できます。

タスク	ステップ
テストまたはキーワード シーケンスを Silk Test で開く	Silk Test で開く をクリックして、選択したテストまたはキーワード シーケンスを Silk Test で開きます。
キーワードの追加	<ol style="list-style-type: none">1. キーワード リストの一番下にある 新しいキーワード をクリックするか、キーワードを右クリックして、コンテキスト メニューから キーワードを上挿入 を選択します。  注: キーワードの使用状況に基づいて、Silk Test にキーワードを推薦させることができます。推薦させるかどうかを切り替えるには、コンテキスト メニューの レコメンドの有効化 または レコメンドの無効化 を使用します。詳細については、「Silk Test が推薦するキーワード」を参照してください。2. 利用可能なキーワードのリストからキーワードを選択するか、新しいキーワードを作成します。3. 保存 をクリックします。 または、右側にある すべてのキーワード ペインから、既存のキーワードをダブルクリックするか、ドラッグ&ドロップします。  ヒント: Ctrl + クリック を使用すると、複数のキーワードを選択できます。ドロップするとき、キーワードは選択した順番に並び替えられます。
キーワードの削除	削除するキーワードの アクション 列で  をクリックします。 保存 をクリックします。
キーワードの順序の変更	キーワードを移動したい位置にドラッグ&ドロップします。 保存 をクリックします。
キーワード シーケンス (他のキーワードから構成されるキーワード) の作成	<ol style="list-style-type: none">1. キーワード リストから結合したいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。2. 選択範囲を右クリックして、結合 をクリックします。3. 新しいキーワード シーケンスの 名前 と 説明 を指定します。
キーワード シーケンスからキーワードの抽出	キーワード シーケンスを右クリックし、 キーワードの抽出 を選択します。これによって、元のキーワード シーケンスがそれに含まれるキーワードによって置換されますが、ライブラリからは削除されません。 保存 をクリックします。

タスク	ステップ
テストまたはキーワード シーケンスへのキーワードのコピーと貼り付け	<ol style="list-style-type: none"> 1. キーワード リストからコピーしたいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。 2. Ctrl + C を押すと選択項目がコピーされます。キーワードを移動する場合には、Ctrl + X を押します。 3. キーワードをコピーするテストまたはキーワード シーケンスを開き、キーワードを挿入する上の行を選択します。 4. Ctrl + V を押します。 <p> ヒント: 選択したキーワードを Excel に貼り付けて編集し、それをコピーしてテストまたはキーワード シーケンスに貼り付けることもできます。</p>
キーワード シーケンスのパラメータの定義	<ol style="list-style-type: none"> 1. キーワード リスト上の パラメータ をクリックします。パラメータ ダイアログ ボックスが表示されます。 2. パラメータの追加 をクリックします。 3. 新しいパラメータに対して 名前 を指定します。パラメータが出力パラメータ (入力値を必要とするのではなく、値を戻します) の場合、出力 チェック ボックスをチェックします。 4. OK をクリックします。 5. 保存 をクリックします。
ドラフト キーワードの編集	<ol style="list-style-type: none"> 1. 編集するドラフト キーワードのアクション 列で、 をクリックします。 2. キーワードの グループ を選択するか、新しいグループを指定します。 3. 説明 にキーワードの説明を入力します。この情報は、キーワードを実装するエンジニアにとって有益な情報です。 4. OK をクリックします。 5. 省略可能 : パラメータ フィールドをクリックして、キーワードのパラメータを追加します。キーワードが Silk Test で実装されると、これらのパラメータは、生成されたコード スタブに現れます。 6. 保存 をクリックします。
キーワードの検索	<p>右側にある すべてのキーワード ペインの検索フィールドを使用して、特定のキーワードを検索します。文字を入力するとリストが更新され、その文字に一致する既存のキーワードが動的に表示されます。検索のヒント :</p> <ul style="list-style-type: none"> • 検索では大文字小文字は区別されません。doAction を入力すると、doaction や DOAction が表示されます。 • 大文字のみを入力すると、いわゆる キャメルケース検索が実行されます。ECD を入力すると、Enter Car Details、Enter Contact Details、EnterContactDetails が表示されます。 • キーワード名とグループ名が検索対象になります。test を入力すると、test を含むすべてのキーワードと、グループ名に test を含むグループのすべてのキーワードが表示されます。 • ? は、0 または 1 文字と一致します。user?test を入力すると、userTest や usersTest が表示されます。 • * は、0 または n 文字と一致します。my*keyword を入力すると、myKeyword や myNewKeyword や my_other_keyword が表示されます。 • <文字列>. はグループ名のみを検索します。group. を入力すると、グループ名に group を含むグループのすべてのキーワードが表示されます。

タスク	ステップ
	<ul style="list-style-type: none"> • .<文字列> はキーワード名のみを検索します。keyword を入力すると、keyword を含むすべてのキーワードが表示されます。 • <string>.<string> は特定のグループのキーワードを検索します。group.word を入力すると、myGroup グループの myKeyword が表示されます。 • 引用符を使用して完全一致のみを検索します。'Keyword' を入力すると、Keyword や MyKeyword は表示されますが、keyword は表示されません。

Silk Test のキーワード レコメンド機能

キーワード駆動テスト エディター でキーワード駆動テストにキーワードを追加するとき、Silk Test は、そのテストの次のキーワードとして使用する可能性のある既存のキーワードを推薦します。推薦するキーワードはキーワード リストの上位にリストされ、Silk Test がどの程度そのキーワードを推薦しているかが、棒グラフの塗りつぶした長さによって示されます。

Silk Test は、次のアルゴリズムに基づいてキーワードを推薦します。

- キーワード駆動テストまたはキーワード シーケンスに最初のキーワードを追加する場合、Silk Test は他のキーワード駆動テストまたはキーワード シーケンスの最初のキーワードとして使用されていると同様なキーワードを検索します。最も頻繁に使用されるキーワードがより上位に推薦されます。
- 既に他のキーワードを含んでいるキーワード駆動テストまたはキーワード シーケンスに、さらにキーワードを追加する場合、Silk Test は次のようにしてキーワードを推薦します。
 - 新しいキーワードを追加するキーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがある場合、Silk Test は前のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせに続いて現れるキーワードを推薦します。
 - キーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがないが、現在の位置の後にキーワードがある場合、Silk Test は後のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせの前に現れるキーワードを推薦します。
- さらに、Silk Test は見つかったキーワードの類似性も考慮します。たとえば、2 つのキーワードの名前とグループの両方が一致する場合、Silk Test は名前だけが一致するキーワードよりもこれらをより上位に推薦します。
- Silk Central との接続が確立すると、現在のプロジェクトに対応するキーワード ライブラリに属したキーワード駆動テストに含まれるキーワードも考慮されます。

キーワードでのパラメータの使用

キーワードまたはキーワード シーケンスは、入力パラメータと出力パラメータを持つことができます。このトピックでは、Silk Test を使用してこれらのパラメータを処理する方法について説明します。

キーワード駆動テスト エディター で、キーワードまたはキーワード シーケンスに対して定義されたパラメータを表示し、パラメータの値を編集できます。

キーワード ウィンドウ で、キーワードまたはキーワード シーケンス上にマウス カーソルを動かすと、キーワードまたはキーワード シーケンスに割り当てられているパラメータを確認できます。

単純なキーワードの入力パラメータ

他のテストメソッドに対するものと同じ方法で、キーワードの入力パラメータを定義して使用することができます。

次のサンプルコードでは、2つの入力パラメータ `userName` と `password` を持つキーワード `setUserDetails` を定義する方法を説明します。

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

単純なキーワードの出カパラメータ

キーワードの戻り値または1つ以上の出力パラメータを定義することができます。また、戻り値と1つ以上の出力パラメータを組み合わせて使用することもできます。

次のサンプルコードでは、文字列を返すキーワード `getText` を定義する方法を説明します。

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

次のサンプルコードでは、2つの出力パラメータ `userName` と `password` を持つキーワード `getUserDetails` を定義する方法を説明します。

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

キーワードシーケンスのパラメータ

キーワードシーケンスのパラメータは、**パラメータ** ダイアログボックスで定義し、編集できます。このダイアログボックスは、**キーワードシーケンスエディター**の**パラメータ**をクリックして開くことができます。

例：パラメータを取るキーワード

このトピックでは、パラメータを取るキーワードを使用する方法の例を紹介します。

最初の手順として、使用するキーワードを含んだキーワード駆動テストを作成します。これは、キーワード駆動テスト全体を記録するか、新しいキーワード駆動テストを作成してからキーワード駆動テストエディターでキーワードを追加することによって行います。

この例では、キーワード駆動テストは次のつのキーワードを含んでいます。

- Start application** これは、AUTを開始して基本状態を設定する標準キーワードです。
- Login** これは、ユーザー名とパスワードで識別される特定のユーザーでAUTにログインするキーワードです。
- GetCurrentUser** このキーワードは、AUTに現在ログインしているユーザーの名前を返します。
- AssertEquals** このキーワードは、2つの値を比較します。
- Logout** これは、AUTからユーザーをログアウトするキーワードです。

次のステップでは、パラメータをキーワードに追加します。これを行うには、キーワードのテストスクリプトを開き、メソッドにパラメータを追加します。

入力パラメータ UserName と Password をキーワード Login に追加するには、

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

を、次のように変更します。

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







出力パラメータ UserName をキーワード GetCurrentUser に追加するには、

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










を、次のように変更します。

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

キーワード駆動テスト エディター 上のキーワード駆動テストは、次のようになります。

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

これで、**キーワード駆動テスト エディター** 上で入力パラメータの実際の値を指定できます。キーワード GetCurrentUser の出力パラメータ UserName の値を取得するには、`${current user}` のように、変数を指定します。その後のキーワードで、変数を指定して格納された値を渡すことができます。


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

Silk Test でキーワード駆動テストを作成する

Silk Test でキーワード駆動テストを作成する前に、プロジェクトを選択する必要があります。

キーワード駆動テスト エディター を使って、新しいキーワードと既存のキーワードを新しいキーワード駆動テストに結合できます。新しいキーワードは、後のステップで自動テストのメソッドとして実装する必要があります。

このトピックでは、Silk4J でキーワード駆動テストを作成する方法について説明します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。

1. **Silk4J** > **新規キーワード駆動テスト** をクリックします。 **新規キーワード駆動テスト** ダイアログ ボックスが開きます。
 2. 新しいテストの名前を **名前** フィールドに入力します。
 3. 新しいテストを追加したいプロジェクトを選択します。
デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しいテストが作成されます。
-  **注:** Silk Test が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。
4. **終了** をクリックします。
 5. **いいえ** をクリックして、空のキーワード駆動テストを作成します。 **キーワード駆動テスト エディター** が開きます。
 6. 次のアクションのいずれかを実行します。
 - 新しいキーワードを追加する場合は、 **新しいキーワード** フィールドにキーワードの名前を入力します。
 - 既存のキーワードを追加する場合は、リストを展開して追加するキーワードを選択します。
 7. Enter を押します。
 8. 実行するすべてのキーワードを追加するまで、上記の 2 つの手順を繰り返します。
 9. **保存** をクリックします。

続いて、キーワードを実装します。すべてのキーワードが実装されている場合は、テストを実行します。


Silk Test でのキーワード駆動テストの記録


Silk Test でキーワード駆動テストを作成する前に、プロジェクトを選択する必要があります。

単一のキーワードを記録する場合は、「[キーワードの記録](#)」を参照してください。

キーワード駆動テストを記録するには：

1. **Silk4J** > **新規キーワード駆動テスト** をクリックします。 **新規キーワード駆動テスト** ダイアログ ボックスが開きます。
2. 新しいテストの名前を **名前** フィールドに入力します。
3. 新しいテストを追加したいプロジェクトを選択します。
デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しいテストが作成されます。

 **注:** Silk Test が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

4. **終了** をクリックします。
 5. **はい** をクリックして、キーワード駆動テストの記録を開始します。
 6. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザーの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
 7. 開いているダイアログに応じて、次のいずれかを実行します。
 - **アプリケーションの選択** ダイアログ ボックスで、**OK** をクリックします。
 - **ブラウザーの選択** ダイアログ ボックスで、**記録** をクリックします。
 8. テスト対象アプリケーションで、最初のキーワードに含める操作を実行します。
記録中に利用可能な操作についての詳細は、Silk Test クライアントのドキュメントを参照してください。
 9. キーワードの名前を指定するには、**記録中** ウィンドウでキーワードの名前の上にマウス カーソルを動かして、**編集** をクリックします。
 **注:** Silk Test は、キーワード駆動テストの開始にアプリケーションの開始 キーワードを自動的に追加します。このキーワードで、アプリケーションの基本状態が実行され、テストを正しく再生できるようにになります。基本状態についての詳細は、Silk Test クライアントのドキュメントを参照してください。
 - 10 キーワードの名前を **キーワードの名前** フィールドに入力します。
 - 11 **OK** をクリックします。
 - 12 次のキーワードの操作を記録するには、**新しいキーワードの名前** フィールドに新しいキーワードの名前を入力し、**追加** をクリックします。Silk Test は、新しいキーワードに新しい操作を記録します。
 - 13 キーワード駆動テスト全体を記録すまで、新しいキーワードを作成し、キーワードに対する操作を記録します。
 - 14 **停止** をクリックします。**記録完了** ダイアログ ボックスが開きます。
- Silk Test は、すべての記録したキーワードを含む新しいキーワード駆動テストを作成します。

Silk Test でのキーワード駆動テストの基本状態の設定

Silk Test でキーワード駆動テストを実行すると、キーワード駆動テストは基本状態のキーワードを呼び出すことにより、Silk Test は AUT を基本状態から開始します。

キーワード駆動テストの記録時に、Silk Test は基本状態のキーワード (isBaseState プロパティが *true* に設定されているキーワード) を現在のプロジェクトから検索します。

- 基本状態のキーワードが現在のプロジェクトに存在した場合、Silk Test は、キーワード駆動テストの最初のキーワードとして、このキーワードを挿入します。
- 基本状態のキーワードがプロジェクトに存在しない場合、Silk Test は、アプリケーションの開始 という名前の新しい基本状態のキーワードを作成し、キーワード駆動テストの最初のキーワードとして挿入します。

キーワードを手動で基本状態のキーワードとしてマークするには、isBaseState プロパティを Keyword アノテーションに追加し、プロパティの値を *true* に設定します。

```
@Keyword(value = "Start application", isBaseState = true)
public void start_application() {
```

```
// Base state implementation  
}
```

Silk Test でのキーワードの実装

キーワードを実装する前に、キーワード駆動テストの一部としてキーワードを定義します。

次の手順は、Silk4J でキーワードを実装する方法を示します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。

キーワード駆動テストで再利用するためにキーワードを実装するには：

1. 実装するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。**キーワードの場所の選択** ダイアログ ボックスが開きます。
3. **選択** をクリックして、キーワード実装に追加するパッケージやクラスを選択します。
4. 省略可能：**パッケージ** フィールドに、新しいキーワード実装のパッケージ名を入力します。
5. **クラス** フィールドに、新しいキーワード実装のクラス名を入力します。
6. **OK** をクリックします。
7. 次のアクションのいずれかを実行します。
 - キーワードを記録するには、**はい** をクリックします。
 - 空のキーワード メソッドを作成するには、**いいえ** をクリックします。
8. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザーの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
9. **記録** をクリックします。
記録についての詳細は、Silk Test クライアントのドキュメントを参照してください。

実装したキーワードが **キーワード** ウィンドウで未実装として表示されている場合は、Eclipse メニューで **プロジェクト > 自動的にビルド** をチェックします。

Silk Test での Silk Central キーワードの実装

Silk Central キーワードを実装する前に、Silk Central でキーワード駆動テストの一部としてキーワードを定義します。

次の手順は、Silk4J で Silk Central キーワードを実装する方法を示します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。

Silk Test で Silk Central キーワードを実装するには：

1. Silk Test のプロジェクトをキーワード駆動テストを含んだ Silk Central のキーワード ライブラリと同じ名前で作成します。

2. Silk Central のキーワード ライブラリにタイプが割り当てられていない場合、**Silk4J > キーワード ライブラリのアップロード** をクリックして、ライブラリ タイプを設定します。
3. 省略可能 : Silk Central の特定のキーワードを Silk Test で実装するには、ライブラリの **キーワード タブ** を Silk Central で開き、キーワードの **アクション** 列から **Silk Test で実装** をクリックします。
4. Silk Test メニューで、**Silk4J > キーワード ビューの表示** をクリックします。
5. **キーワード ビュー** で、キーワード駆動テストをダブルクリックします。
キーワード ビュー で Silk Central による変更を更新するには、**更新** をクリックします。
6. ツールバーで、**操作の記録** をクリックします。
7. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
8. **記録** をクリックします。
記録についての詳細は、Silk Test クライアントのドキュメントを参照してください。
9. 最初の未実装のキーワードの操作を記録します。
- 10 現在のキーワードに対するすべての操作を記録し終わったら、**次のキーワード** をクリックします。
- 11 **記録中** ウィンドウでキーワードを切り替えるには、**前のキーワード** と **次のキーワード** をクリックします。
- 12 **停止** をクリックします。**記録完了** ダイアログ ボックスが開きます。



注: Silk Central のキーワード駆動テストは、Silk Test では読み取り専用であるため、キーワードを削除したり、キーワードの順番を変更することはできません。

実装したキーワードが **キーワード** ウィンドウで未実装として表示されている場合は、Eclipse メニューで **プロジェクト > 自動的にビルド** をチェックします。

Silk Test でのキーワードの記録

完全に新しいキーワードに対してではなく、キーワード駆動テストに既に存在するキーワードに対しては、操作の記録のみを行えます。新しいキーワード駆動テストを記録する場合は、「[キーワード駆動テストの記録](#)」を参照してください。

新しいキーワードの操作を記録するには :

1. 記録するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。**キーワードの場所の選択** ダイアログ ボックスが開きます。
3. **選択** をクリックして、キーワード実装に追加するパッケージやクラスを選択します。
4. 省略可能 : **パッケージ** フィールドに、新しいキーワード実装のパッケージ名を入力します。
5. **クラス** フィールドに、新しいキーワード実装のクラス名を入力します。
6. **OK** をクリックします。
7. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザの選択** ダイアログ ボックスが開きます。



- a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
8. **記録** をクリックします。記録中 ウィンドウが開き、Silk Test はキーワードの操作の記録を開始します。
9. テスト対象アプリケーションで、テストする操作を実行します。
記録中に利用可能な操作についての詳細は、Silk Test クライアントのドキュメントを参照してください。
- 10 **停止** をクリックします。記録完了 ダイアログ ボックスが開きます。
- 記録した操作は、定義したクラスのコンテキストに表示されます。

キーワード駆動テストの編集



注: Silk Test では、Silk Test にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テスト エディター** でキーワード駆動テストを開き、**編集** をクリックします。

キーワード駆動テストを編集するには：

1. **キーワード駆動テスト エディター** でキーワードを開きます。
 - a) Silk Central メニューから、**テスト > 詳細ビュー** を開き、キーワード駆動テストが存在するプロジェクトを展開します。
 - b) **テスト** ツリーでキーワード駆動テストを選択します。
 - c) **キーワード** タブを選択します。
2. 新しいキーワードをキーワード駆動テストに追加するには：
 - a) **新しいキーワード** フィールドをクリックします。
 - b) 新しいキーワードの名前を入力します。
 - c) Enter を押します。
3. 既存のキーワードを編集するには、キーワードの左側にある **キーワードを開く** をクリックします。
 -  **注:** Silk Central は、Silk Central で作成したすべてのキーワードの所有権を持ちます。このことは、このようなキーワードに対して行う変更は Silk Test ではなく、Silk Central で保存されることを意味します。
4. キーワードをキーワード駆動テストにコピーするには：
 - a) キーワードを選択します。
 -  **ヒント:** 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。
 - b) **Ctrl + C** を押します。
 - c) キーワードを挿入する上の行を選択します。
 - d) **Ctrl + V** を押します。
5. キーワード駆動テストのほかの場所にキーワードを移動するには、キーワードをクリックして他の場所にドラッグするか、または次の手順を実行します。
 - a) キーワードを選択します。



ヒント: 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。

- b) **Ctrl + X** を押します。
 - c) キーワードを挿入する上の行を選択します。
 - d) **Ctrl + V** を押します。
6. キーワード駆動テストからキーワードを削除するには、キーワードの左側にある **キーワードの削除** をクリックします。
キーワードは、**キーワード** ウィンドウでまだ利用可能なので、いつでもキーワード駆動テストに再度追加することができます。
7. 変更を保存するには、**保存** をクリックします。

キーワードのキーワード シーケンスへの結合

キーワード駆動テスト エディター を使って、複数のキーワード駆動テストで順番に実行したいキーワードを結合してキーワード シーケンスを作成できます。

1. 結合するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、Ctrl キーを押しながら結合したいキーワードをクリックします。
3. 選択範囲を右クリックして、**結合** をクリックします。**キーワードの結合** ダイアログ ボックスが開きます。
4. **名前** フィールドに、新しいキーワード シーケンスの名前を入力します。
5. 省略可能：**説明** フィールドに、新しいキーワード シーケンスの説明を入力します。
6. **結合** をクリックします。

新しいキーワード シーケンスが開き、**キーワード** ウィンドウにも表示されます。キーワード駆動テストでキーワード シーケンスを使用できます。




注: 他のキーワードと同じように、キーワード シーケンス自身を実行することはできませんが、キーワード駆動テストの一部として実行することができます。

Eclipse からのキーワード駆動テストの再生

次の手順は、Silk4J でキーワード駆動テストを再生する方法を示します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。

1. **プロジェクト エクスプローラー** で、再生するキーワード駆動テスト資産に移動します。
2. 資産名を右クリックします。
3. **実行 > キーワード駆動テスト** を選択します。
4. 省略可能：**構成の実行** ダイアログ ボックスを開くには、**実行 > 実行の構成** を選択します。
5. 省略可能：**実行構成** ダイアログ ボックスで、他のテストやプロジェクトを選択できます。
6. 省略可能：**実行の構成** ダイアログ ボックスの **グローバル変数** グリッドで、キーワード駆動テストの実行に使用される任意の変数の値を設定できます。これらの値は、キーワード駆動テスト資産を実行するときに常に使用されます。
 - a) 変数の **変数名** と **値** を対応するフィールドに入力します。
 - b) **Enter** を入力すると、グリッドに新しい行が追加されます。
 - c) 前の 2 つの手順を繰り返して、使用するすべてのグローバル変数の値を設定します。

Silk Central などのテスト管理ツールで管理されている自動化フレームワークの一部としてキーワード駆動テストを実行する場合、新しい `.properties` ファイルをプロジェクトに追加して、プロジェクト全体のグローバル変数の値を設定できます。詳細については、「変数を指定したキーワード駆動テストの再生」を参照してください。

7. 省略可能： **実行の構成** ダイアログ ボックスを閉じて、キーワード駆動テスト資産の実行を開始するには、**実行** をクリックします。
8. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。
 **注:** 複数のアプリケーションが現在のに対して設定されている場合、**ブラウザーの選択** ダイアログ ボックスは表示されません。
9. 省略可能：必要に応じて、両方の **Shift** キーを同時に押して、テストの実行を停止できます。
10. テストの実行が完了すると、**再生完了** ダイアログ ボックスが開きます。**結果の検討** をクリックして、完了したテストの TrueLog を確認します。

Silk Central からの Silk Test テストの再生

Silk Central から Silk Test テストにアクセスするには、Silk Central がソース管理プロファイルを介してアクセスできるリポジトリに Silk Test テストを含んだ JAR ファイルを格納する必要があります。

Silk Central から Silk Test の機能テスト (キーワード駆動テストなど) を再生するには：

1. Silk Central で、Silk Test テストを実行するプロジェクトを作成します。
2. **テスト > 詳細ビュー** を開き、新しいプロジェクト用に新しいテスト コンテナを作成します。

Silk Central に関する追加の情報については、『[Silk Central ヘルプ](#)』を参照してください。

テスト コンテナには、Silk Test テストを格納するソース管理プロファイルを指定する必要があります。

- a) **テスト ツリー**で、その下に新しいテスト コンテナを追加するノードをクリックします。
 - b) **テスト コンテナの新規作成** をクリックします。**テスト コンテナの新規作成** ダイアログ ボックスが開きます。
 - c) **名前** フィールドに、新しいテスト コンテナの名前を入力します。
たとえば、キーワード駆動テストを入力します。
 - d) **ソース管理プロファイル** フィールドに、Silk Test テストを含んだ JAR ファイルを格納するソース管理プロファイルを選択します。
 - e) **OK** をクリックします。
3. 新しいテスト コンテナに新しい JUnit テストを作成します。

Silk Central に関する追加の情報については、『[Silk Central ヘルプ](#)』を参照してください。

- a) **JUnit テスト プロパティ** ダイアログ ボックスの **テスト クラス** フィールドに、テスト クラスの名前を入力します。

テストスイート クラスの完全修飾名を指定します。詳細については、「[コマンド ラインからのキーワード駆動テストの再生](#)」を参照してください。

- b) **クラスパス** フィールドに、テストを含む JAR ファイルの名前を指定します。
- c) キーワード駆動テストの場合、セミコロンで区切って次のファイルへのパスも指定します。

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

これらのファイルは、Silk Test インストール ディレクトリにあります。たとえば、キーワード駆動テストが tests.jar JAR ファイルに含まれている場合、**クラスパス** フィールドは、次のようになります。

```
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. 終了 をクリックします。

5. テストを実行します。

Silk Central でのテストの実行に関する追加の情報については、『[Silk Central ヘルプ](#)』を参照してください。

コマンド ラインからのキーワード駆動テストの再生

このタスクを実行する前に、JDK の場所を参照できるように PATH 変数を更新する必要があります。詳細については、『[JDK Installation for Microsoft Windows](#)』を参照してください。

CI サーバーからテストを再生する場合など、コマンド ラインからキーワード駆動テストを再生するには、KeywordTestSuite クラスを使用します。

1. コマンド ラインからキーワード駆動テストを実行するには、@KeywordTests アノテーションを使用して JUnit テスト スイートを作成します。たとえば、キーワード駆動テスト *My Keyword-Driven Test* を実行する場合は、次のような JUnit テスト スイート *MyTestSuite* を作成します。

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

2. CLASSPATH に以下を含めます。


- junit.jar
- org.hamcrest.core JAR ファイル
- silktest-jtf-nodeps.jar
- com.borland.silk.keyworddriven.engine.jar
- キーワード駆動テストを含んだフォルダの 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. 省略可能：新しい *.properties* ファイルをプロジェクトに追加して、キーワード駆動テストの実行に使用する任意の変数の値を設定します。

詳細については、「[変数を指定したキーワード駆動テストの再生](#)」を参照してください。

4. `java org.junit.runner.JUnitCore <Name>` を入力して JUnit テスト メソッドを実行します。ここで、*Name* は、最初の手順で作成した JUnit テスト スイートの名前です。

 **注:** トラブルシューティングの情報については、次の JUnit のドキュメントを参照してください：
http://junit.sourceforge.net/doc/faq/faq.htm#running_1.

例

たとえば、*My Keyword Driven Test 1* と *My Keyword Driven Test 2* の 2 つのキーワード駆動テストを実行するには、次のクラスを作成します。

```
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 {  
  
}
```

コマンドラインからクラスを実行するには、次のように入力します。

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

ファイル `c:%temp%globalvariables.properties` に格納されたグローバル変数を使用してコマンドラインからクラスを実行するには、次のように入力します。

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

詳細については、「変数を指定したキーワード駆動テストの再生」を参照してください。

Apache Ant を使用したキーワード駆動テストの再生

このトピックで述べる手順を実行するには、コンピュータに Apache Ant がインストールされている必要があります。

Apache Ant を使用してキーワード駆動テストを再生し、たとえば、テスト実行の HTML レポートを生成するには、`KeywordTestSuite` クラスを使用します。

1. Apache Ant を使用してキーワード駆動テストを実行するには、`@KeywordTests` アノテーションを使用して JUnit テストスイートを作成します。たとえば、キーワード駆動テスト *My Keyword-Driven Test* を実行する場合は、次のような JUnit テストスイート `MyTestSuite` を作成します。

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

2. キーワード駆動テストを含んだ Silk Test プロジェクトの `build.xml` ファイルを開きます。
3. キーワード駆動テストを実行するには、次のターゲットを `build.xml` ファイルに追加します。

```
<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>
```

JUnit タスクの詳細については、『<https://ant.apache.org/manual/Tasks/junit.html>』を参照してください。

4. 省略可能：すべてのテストの XML レポートを作成するには、ターゲットに次のコードを追加します。

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


5. 省略可能 : XML レポートから HTML レポートを作成するには、ターゲットに次のコードを追加します。

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

JUnitReport タスクの詳細については、『<https://ant.apache.org/manual/Tasks/junitreport.html>』を参照してください。

完全なターゲットは、次のようになります。

```
<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. Eclipse からテストを実行するには、以下の手順を実行します。

- パッケージ・エクスプローラーで、build.xml ファイルを右クリックします。
- 実行 > Ant ビルド... を選択します。
- 構成の編集 ダイアログ ボックスの **ターゲット** タブで、**runTests** をチェックします。
- 実行 をクリックします。

コマンドラインや CI サーバーからテストを実行することもできます。詳細については、『<https://ant.apache.org/manual/running.html>』および Silk Test ヘルプ の「CI (継続的インテグレーション) サーバーからのテストの再生」を参照してください。

変数を指定したキーワード駆動テストの再生

キーワード駆動テスト用の変数の値を設定する前に、プロジェクトを作成する必要があります。

ユーザーが実行するキーワード駆動テスト資産のすべての実行に対するグローバル変数の値を設定するには、**実行の構成** ダイアログ ボックスの **グローバル変数** グリッドを使用します。詳細については、「Eclipse からのキーワード駆動テストの再生」を参照してください。

Silk Central などのテスト管理ツールで管理されている自動化フレームワークの一部としてキーワード駆動テストを実行するときに、Silk Test でのキーワード駆動テストの実行に使用する変数の値を設定できま

す。プロジェクト全体に対するグローバル変数の値 (つまり、このプロジェクトのキーワード駆動テスト資産を Silk Test ユーザーが実行するときに常にこれらの値が使用されます) を設定するには、次の手順を実行します。

1. **パッケージ・エクスプローラー** で、変数を指定して実行するキーワード駆動テストが存在するプロジェクトを展開します。
2. プロジェクトの **src** フォルダを右クリックして、**新規 > ファイル** を選択します。 **新規ファイル** ダイアログ ボックスが開きます。
3. **ファイル名** フィールドに、`globalvariables.properties` を入力します。
4. **終了** をクリックします。新しいプロパティ ファイルが開きます。
5. ファイルの新しい行を追加して変数を指定します。

新しい変数のフォーマットは次のようになります。

```
name=value
```

たとえば、`user` と `password` という 2 つの変数を指定する場合は、次のように入力します。

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

プロパティ ファイルのフォーマットや、空白類などの Unicode 文字の入力方法についての情報は、『[Properties File Format](#)』を参照してください。

6. `globalvariables.properties` ファイルを保存します。

プロジェクトのキーワード駆動テストが Silk Test から実行されるときはいつでも、変数が使用されます。

キーワードのグループ化

キーワードをライブラリで構造化するために、グループ化することができます。

このトピックでは、Silk4J でキーワードを特定のグループに追加する方法について説明します。Silk4NET での手順と Silk Test Workbench での手順はほとんど同じです。特定の Silk Test クライアントを使用してキーワード駆動テストを実行する手順についての詳細は、Silk Test クライアントのドキュメントを参照してください。このグループ名は Silk Central でも使用され、キーワードはグループ名に従って分類されます。

キーワードを特定のグループに追加するには：

1. キーワードの実装を開きます。
 - a) キーワードを実装しているプロジェクトを開きます。
 - b) **キーワード** ウィンドウを開きます。
 - c) **キーワード** ウィンドウで、キーワードを選択します。
 - d) **実装へ移動** をクリックします。
2. クラスのすべてのメソッドをキーワード グループに追加するには、クラス定義の前にキーワード グループを追加します。

たとえば、キーワードを `Calculator` グループ追加するには、次のように入力します。

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

キーワード ウィンドウで表示されるキーワード名にグループが含まれるようになります。たとえば、`Addition` キーワードが `Calculator` グループに属している場合は、`Calculator.Addition` として表示されません。

索引

A

Ant

キーワード駆動テストの実行 24

S

Silk Central

キーワードのアップロード 7
テストの実行 22
場所の設定 6

Silk Central キーワード

実装 18

あ

アップロードする

キーワードライブラリ 9
ライブラリ 9

アプリケーション構成

キーワード駆動テスト 17

か

概念

テスト自動化 4

管理

キーワード 11

き

キーワード

Silk Central にアップロードする 7
管理 11
記録 19
グループ化 26
結合 11, 21
削除 11
シーケンス 11
実装 11, 18
置換 11
追加 11
について 5
ネスト 11
パラメータ 11, 13
パラメータ、例 14
開く 11

キーワードシーケンス

作成 21
パラメータ 13

キーワードライブラリ

アップロードする 9

キーワード駆動

テスト 4

キーワード駆動テスト

Ant での実行 24

Silk Central から実行 22

Silk Central キーワードの実装 18

アプリケーション構成 17

概要 5

キーワード レコメンド、アルゴリズム 13

キーワードのアップロード、Silk Central 7

キーワードの削除 20

キーワードの実装 18

キーワードの追加 20

基本状態 17

記録 16

コマンド ラインからの実行 23

再生 21

作成 16

作成する 10

手動テストから変換する 10

パラメータ、例 14

編集 20

変数を指定した実行 25

利点 5

停止中 21

キーワード駆動テスト エディター

推薦するキーワード 13

キーワード駆動テストの実行

変数 25

キーワードの削除

キーワード駆動テスト 20

キーワードの追加

キーワード駆動テスト 20

基本状態

キーワード駆動テスト 17

記録

キーワード 19

キーワード駆動テスト 16

く

グループ化

キーワード 26

け

継続的インテグレーション

キーワードライブラリのアップロード 9

結合

キーワード 11

こ

コマンド ライン

キーワード駆動テストの実行 23

さ

削除

キーワード 11

作成
 キーワード駆動テスト 16
作成する
 キーワード駆動テスト 10

し

実装
 キーワード 11
手動テスト
 キーワード駆動テストに変換する 10

す

推薦するキーワード
 キーワード駆動テスト エディター 13

ち

置換
 キーワード 11

つ

追加
 キーワード 11

て

テスト自動化
 概要 4
テストの実行
 Silk Central 22

と

統合

Silk Central の場所の設定 6

ね

ネスト
 キーワード 11

は

パラメータ
 処理、キーワード 13

ひ

開く
 キーワード 11

へ

変換
 手動テストからキーワー駆動テスト 10
変数
 キーワード駆動テストの実行 25

ら

ライブラリ
 アップロードする 9

れ

レコメンド
 アルゴリズム 13