

SilkTest 2011

Silk4NET ユーザー ガイド

Borland[®]
(A MICRO FOCUS COMPANY)

 **MICRO**[®]
FOCUS

Micro Focus
575 Anton Blvd., Suite 510
Costa Mesa, CA 92626

Copyright © Micro Focus IP Development Limited 2011. All rights reserved. SilkTest
は Borland Software Corporation に由来する成果物を含んでいます, Copyright 2011
Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, Micro Focus ロゴ、及びその他は 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。

2011-10-05

目次

Silk4NET	4
Visual Studio での Silk4NET プロジェクトの作成	4
Silk4NET テストの作成の概要	4
Visual Studio プロジェクトへの Silk4NET テスト テンプレートの追加	5
Visual Studio での Silk4NET テストの手動作成	6
テストの記録	6
記録したテスト ステップのドラッグ&ドロップ	7
記録したテスト ステップの Visual Studio へのエクスポート	7
Visual Studio への Silk4NET プロジェクトのエクスポート	8
Silk4NET テストの実行	8
テスト結果の分析	9
Silk4NET サンプル テスト	9

Silk4NET

Microsoft Visual Studio の SilkTest 用のプラグインである Silk4NET を使用すると、機能テスト、回帰テスト、ローカリゼーション テストの作成と管理を Visual Studio で直接、効率的に実行できます。Silk4NET では、Visual Basic .NET または C# のいずれかを使用してテストを開発し、Microsoft のテスト環境におけるテスト計画またはビルド プロセスの一環としてテストを実行し、すべてのテスト結果を Visual Studio 内で表示することができます。

Silk4NET では、AJAX と Web 2.0、RCP、WPF、Windows Forms と Win32 など、広範囲にわたるアプリケーション テクノロジーがサポートされています。Silk4NET は、複雑なテストに適用した場合にも自動化の利点を実現されるように設計されており、開発者が使い慣れた環境に直接、テストの自動化機能を追加するため、テスト アプリケーションに加えられた変更に対応することができます。

また、Silk4NET の強力なテスト フレームワークを使用すると、複数のテスト プロジェクトに対してテストの再利用が促進されるため、より高い ROI を達成できます。テストスイートの構築と保守に要する時間が短縮されるため、QA 担当者はテスト範囲を拡張し、アプリケーション品質を最適化することができます。

Visual Studio での Silk4NET プロジェクトの作成

1. **ファイル > 新規作成 > プロジェクト** をクリックします。
新しいプロジェクト ダイアログ ボックスが表示されます。
2. インストールされたテンプレートで **Visual C#** または **Visual Basic** を展開し、**Silk4NET** をクリックします。
3. プロジェクト名、場所、ソリューション、ソリューション名を入力し、**OK** をクリックします。
Silk4NET プロジェクトを含む新しいソリューションが作成されます。また、このプロジェクトには、言語固有の以下のファイル名を使用して、Silk4NET テストも作成されます。
 - UnitTest1.vb
 - UnitTest1.cs

Silk4NET テストの作成の概要

テストを手動で作成する場合の開始点として、Silk4NET テスト テンプレートを使用できます。

このテンプレートには、文字エンコード クラス、SilkTest の Open Agent API、そして Visual Studio の単体テストのサポートに必要な名前空間が宣言がされています。また、このテンプレートには、汎用のクラスとメソッドも含まれており、ここに、アプリケーションの機能をテストするために設計したテスト ステップを作成できます。

別の方法として、SilkTest Recorder を使用して、テスト アプリケーションに対するユーザー操作を記録することによって、テストの作成を自動化することもできます。

SilkTest Recorder では、テストがニュートラル形式で記録されます。記録したテストは、Visual Basic .NET または C# のいずれかで Visual Studio プロジェクトにエクスポートできます。

また、SilkTest Recorder では、ドラッグ&ドロップ機能やクリップボード機能もサポートされています。これらの機能を使用すると、記録したテスト ステップを Visual Studio の既存の Silk4NET テストにすばやく移すことができます。この方法では、2つのアプローチを混在させて使用できます。つまり、テストのある部分を記録し、別の部分を手動でコーディングすることができます。

Visual Studio プロジェクトへの Silk4NET テスト テンプレートの追加

1. **プロジェクト > 新しい項目の追加** を選択します。
新しい項目の追加 ダイアログ ボックスが表示されます。
2. **インストールされたテンプレート** で、**共通項目 > Silk4NET** を選択します。
中央のペインに Silk4NET テスト テンプレートが表示されます。
3. **Silk4NET テスト** を選択し、**追加** をクリックします。
プロジェクトのデフォルトのプログラム言語に応じて UnitTest1.cs または UnitTest1.vb というデフォルトのファイル名で、Silk4NET テスト テンプレートがプロジェクトに追加されます。

このテンプレートには、SilkTest の Open Agent API と Visual Studio の単体テストのサポートに必要な名前空間が宣言されています。また、以下の例に示すように、特定のテスト ステップを作成できる汎用のクラスとメソッドも含まれています。

```
'Visual Basic .NET
Imports System.Text
Imports Microsoft.VisualStudio.TestTools.UnitTesting
Imports SilkTest.Ntf

<TestClass(>
Public Class UnitTest1

    Private ReadOnly _desktop As Desktop = Agent.Desktop

    <TestMethod(>
    Public Sub TestMethod1()
    End Sub

End Class
```

```
//C#
Using System;
Using Microsoft.VisualStudio.TestTools.UnitTesting;
Using SilkTest.Ntf;

namespace SilkTest.Ntf.SampleScripts.CSharp
{
    [TestClass]
    public class UnitTest1
    {
        private readonly Desktop _desktop = Agent.Desktop;

        [TestMethod]
        public void TestMethod1()
        {
            // Put your test here.
        }
    }
}
```

Visual Studio での Silk4NET テストの手動作成

1. Silk4NET テスト テンプレートをプロジェクトに追加します。プロジェクト > 新しい項目の追加 > **Silk4NET** を選択し、**追加** をクリックします。
Silk4NET テスト テンプレートがプロジェクトに追加されます。
2. オプションとして特定のアプリケーション テクノロジーのコントロールのサポートを追加するには、以下の例に示すように、アプリケーション テクノロジーの名前空間を参照するテストの先頭にインポート ステートメントを含める必要があります。


```
'Visual Basic .NET
Imports SilkTest.Ntf.Wpf
Imports SilkTest.Ntf.XBrowser
Imports SilkTest.Ntf.Win32
```

```
//C#
Using SilkTest.Ntf.Wpf;
Using SilkTest.Ntf.XBrowser;
Using SilkTest.Ntf.Win32;
```


3. テスト アプリケーションの基本状態を構成します。
例：

```
'Visual Basic .NET
Dim baseState = New BaseState("C:¥¥Windows¥¥system32¥¥notepad.exe",
"/Window[@caption='Untitled - Notepad']")
baseState.WorkingDirectory = "%USERPROFILE%"
baseState.Execute()
```

```
//C#
BaseState baseState = new BaseState("C:¥¥Windows¥¥system32¥¥notepad.exe",
"/Window[@caption='Untitled - Notepad']");
baseState.WorkingDirectory = "%USERPROFILE%"; baseState.Execute();
```

 **注:** 基本状態を使用すると、テストするアプリケーションがフォアグラウンドで実行中であることを保証できます。これにより、テストが常に同じアプリケーション状態で開始されることが保証され、信頼性が高まります。基本状態を使用するには、メイン ウィンドウの外観、およびテストするアプリケーションが実行されていない場合のアプリケーションの起動方法を指定する必要があります。基本状態の作成は任意です。ただし、ベスト プラクティスとして、基本状態を作成することをお勧めします。

4. テスト アプリケーションの目的の機能をテストするクラスとメソッドを追加します。

 **注:** SilkTest の SilkTest Recorder クリップボード機能を使用すると、記録したテスト ステップをコピーして、手動で作成した Silk4NET テストにすばやく貼り付けることができます。

テストの記録

1. Visual Studio のメニュー バーで、**Silk4NET > Recorder の開始** を選択します。
SilkTest Recorder が表示されます。
2. 新しいスクリプトを作成し、記録を開始します。
詳細については、『*SilkTest Recorder ヘルプ*』を参照してください。

Recorder ではテスト操作がニュートラル形式で保存されるので、次のいずれかを行います。

- 記録したテスト ステップを既存の Silk4NET テストにドラッグ & ドロップする。デフォルトのプログラム言語に応じて、テスト ステップが Visual Basic .NET または C# に自動的に変換されます。

- 記録したテスト ステップを、既存の Visual Studio プロジェクトに追加できる Silk4NET テストとしてエクスポートする。
- 記録したテスト ステップを新しいテストにエクスポートし、テストを保存する新しい Silk4NET プロジェクトを作成する。


記録したテスト ステップのドラッグ&ドロップ

記録したテスト ステップのドラッグ&ドロップ先となる Silk4NET テストを作成しておく必要があります。

1. Visual Studio で、記録したテスト ステップのドラッグ&ドロップ先となる Silk4NET テストを開きます。
2. Visual Studio のメニュー バーで、**Silk4NET > Recorder の開始** を選択します。SilkTest Recorder が表示されます。
3. **設定 > グローバル設定** を選択します。**設定** ダイアログ ボックスが表示されます。
4. **デフォルトのクライアント** リストで、使用するプログラム言語を選択します。
5. **OK** をクリックします。
6. 新しいテストを作成し、記録を開始します。
詳細については、『*SilkTest Recorder ヘルプ*』を参照してください。記録を停止すると、記録したテスト ステップが **操作** タブにニュートラル形式で表示されます。
7. Silk4NET テストに追加するステップを選択します。
8. テスト ステップを Silk4NET テストにドラッグ&ドロップします。

テスト ステップがデフォルトのプログラム言語に自動的に変換され、テストに貼り付けられます。

記録したテスト ステップの Visual Studio へのエクスポート

1. SilkTest Recorder のメニュー バーで、**ファイル > エクスポート**を選択します。**エクスポート** ウィザードが開きます。
2. **NTF スクリプトとしてエクスポート** をダブルクリックします。**NTF スクリプトとしてエクスポート** ページが開きます。
3. **エクスポート先** リスト ボックスから、以下のいずれかのオプションを選択します。
 - **クリップボード**: 記録したテスト ステップをクリップボードにコピーします。記録したテスト ステップを Visual Studio プロジェクトの既存の Silk4NET テストにコピーして貼り付ける場合に、このオプションを選択します。
 **注:** このオプションを選択する場合、ソースの場所または基本状態を指定する必要はありません。
 - **NTF スクリプト**: 記録したテスト ステップをテストとしてエクスポートして、既存の Visual Studio プロジェクトに追加できます。新しいテストを作成するか、既存のテストを上書きする場合に、このオプションを選択します。
4. **プログラム言語** リスト ボックスから、テストで Visual Basic .NET または C# のいずれを使用するかを指定します。
5. **テストメソッド** テキスト ボックスに、テスト メソッドの名前を指定します。
たとえば、TestAutoInput と入力します。
6. **名前空間** ボックスに、テストのコンテナ名を指定します。
7. **テストクラス** ボックスに、テストが属するクラス名を指定します。
たとえば、AutoTests と入力します。

8. **ソース・フォルダー** ボックスに、テストをエクスポートする場所を指定します。
使用するフォルダをクリックして移動することもできます。
9. エクスポートしたテストに基本状態を含めるには、**基本状態を使用する** チェック ボックスをオンにします。
基本状態を使用すると、テストするアプリケーションがフォアグラウンドで実行中であることを保証できます。これにより、テストが常に同じアプリケーション状態で開始されることが保証され、信頼性が高まります。基本状態を使用するには、メイン ウィンドウの外観、およびテストするアプリケーションが実行されていない場合のアプリケーションの起動方法を指定する必要があります。基本状態の作成は任意です。ただし、ベスト プラクティスとして、基本状態を作成することをお勧めします。
- 10 **終了** をクリックします。
Recorder により Silk4NET テストが作成され、指定の場所またはクリップボードにエクスポートされます。
- 11 エクスポートした Silk4NET プロジェクトを Visual Studio プロジェクトに追加します。Visual Studio のメニュー バーから **プロジェクト > 既存の項目の追加** を選択し、エクスポートしたテストを選択します。

Visual Studio への Silk4NET プロジェクトのエクスポート

1. SilkTest Recorder のメニュー バーで、**ファイル > エクスポート** を選択します。
エクスポート ウィザードが開きます。
2. **Silk4NET プロジェクトとしてエクスポート** をダブルクリックします。
Silk4NET プロジェクトとしてエクスポート ページが開きます。
3. **プログラム言語** リスト ボックスから、プロジェクトで Visual Basic .NET または C# のいずれを使用するかを指定します。
4. **プロジェクトの場所** テキスト ボックスに、プロジェクトをエクスポートする場所を指定します。
省略可能：使用するフォルダをクリックして移動します。
5. **プロジェクト名** テキスト ボックスに、プロジェクト名を指定します。
たとえば、Visual Basic .NET Sample Project と入力します。
6. **名前空間** テキスト ボックスに、プロジェクトのコンテナ名を指定します。
7. **テストクラス** テキスト ボックスには、テストが属するクラス名を指定します。
たとえば、AutoTests と入力します。
8. **テストメソッド** テキスト ボックスに、テストメソッドの名前を指定します。
たとえば、TestAutoInput と入力します。
9. **終了** をクリックします。
SilkTest Recorder によって、記録したテストを含む新しいプロジェクトが作成され、指定された場所にエクスポートされます。この場所から、Visual Studio でプロジェクトを開くことができます。

Silk4NET テストの実行

1. **テスト > ウィンドウ > テストビュー** を選択します。
テストビュー ウィンドウが開き、テストのリストが表示されます。ウィンドウの上部にある **グループ化リスト** から **プロジェクト**、**名前空間**、**テストクラス** などのフィルタ条件を選択すると、プロジェクトのリストをフィルタ処理できます。
2. 実行するテストを選択します。
3. **テストビュー** ウィンドウの **選択範囲の実行** ボタンをクリックします。

テストビュー ウィンドウからテストを実行する以外に、**テスト** メニュー、開いているテストのコンテキスト依存メニュー、**テスト結果** ウィンドウの実行コマンドも使用できます。

テスト結果の分析

1. Silk4NET テストを実行します。
2. Visual Studio のメニュー バーで、**テスト > ウィンドウ > テスト結果** を選択します。
テスト結果 ウィンドウが表示されます。

テスト結果 ウィンドウに、テスト実行および実行ステータスが表示されます。テスト実行の詳細を表示するには、**テスト結果の詳細の表示** を選択します。

Silk4NET サンプル テスト

Silk4NET サンプル テストは、Visual Studio ソリューションにパッケージ化されており、開いて表示したり、SilkTest サンプル アプリケーションに対して実行したりできます。

Silk4NET サンプル テストにアクセスするには、SilkTest サンプル アプリケーションをインストールする必要があります。このアプリケーションは、http://techpubs.borland.com/silk_gauntlet/SilkTest/ からダウンロードできます。サンプル アプリケーションをインストールしたあと、**スタート > すべてのプログラム > Silk > SilkTest 2011 > Samples > Silk4NET Samples** を選択して、Silk4NET サンプル テストの Visual Studio ソリューション ファイル (Silk4NET Samples.sln) が含まれているフォルダを開きます。

インストールした Silk4NET サンプル アプリケーションに加え、一連の Silk4NET サンプル テストには、Web ベースの以下の SilkTest サンプル アプリケーション用のテストがいくつか含まれています。

Insurance Co. Web サイト <http://demo.borland.com/InsuranceWebExtJS/>

Green Mountain Outpost Web <http://demo.borland.com/gmopost/>