

## SilkTest 13.0

---



Silk4NET ユーザー  
ガイド

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

Copyright © 2012 Micro Focus. All rights reserved. SilkTest は Borland Software Corporation に由来する成果物を含んでいます, Copyright © 2012 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。

2012-04-27

# 目次

<b>Micro Focus へのお問い合わせ</b> .....	4
Micro Focus SupportLine で必要な情報 .....	4
<b>製品通知サービス</b> .....	5
<b>Silk4NET</b> .....	6
<b>SilkTest 製品スイート</b> .....	7
<b>Visual Studio での Silk4NET プロジェクトの作成</b> .....	8
<b>Silk4NET テストの作成の概要</b> .....	9
<b>Visual Studio プロジェクトへの Silk4NET テスト テンプレートの追加</b> .....	10
<b>Visual Studio での Silk4NET テストの手動作成</b> .....	11
<b>テストの記録</b> .....	12
<b>記録したテスト ステップのドラッグ&amp;ドロップ</b> .....	13
<b>記録したテスト ステップの Visual Studio へのエクスポート</b> .....	14
<b>Visual Studio への Silk4NET プロジェクトのエクスポート</b> .....	15
<b>Silk4NET テストの実行</b> .....	16
<b>テスト結果の分析</b> .....	17
<b>Silk4NET サンプル テスト</b> .....	18
<b>動的オブジェクト解決</b> .....	19
XPath の基本概念 .....	19
サポートする XPath のサブセット .....	20
XPath のサンプル .....	21
XPath のパフォーマンス問題のトラブルシューティング .....	22
<b>xBrowser のよくある質問</b> .....	24
要素のテキストに使用されるフォント タイプの確認方法 .....	24
textContent、innerText、および innerHtml の違い .....	24
innerText をカスタム クラス属性として構成したが、ロケータで使えない .....	25
クロス ブラウザ スクリプトの作成時に必要な処置 .....	25
スクリプトで Firefox と Internet Explorer を区別する方法 .....	25
安定したクロス ブラウザ テストを実現するために最適なロケータ .....	25
アプリケーションのログ出力に正しくないタイムスタンプが含まれる .....	25
新しいページに移動したあと、スクリプトがハングする .....	26
正しくないロケータが記録されている .....	26
Windows Internet Explorer で要素を囲む四角形の位置が正しくない .....	26
Link.Select で、Internet Explorer で新しく開いたウィンドウにフォーカスが設定されない .....	26
DomClick(x, y) が Click(x, y) のように動作しない .....	26
FileInputField.DomClick() でダイアログが開かない .....	26
マウス移動設定がオンになっているにもかかわらず、すべての操作が記録されない理由 .....	26
xBrowser API で公開されていない機能が必要な場合の対処方法 .....	27
ロケータでクラスとスタイルの属性が使用されない理由 .....	27
再生中にダイアログが認識されない .....	27

# Micro Focus へのお問い合わせ

Micro Focus は、世界的規模のテクニカル サポートおよびコンサルティング サービスを提供します。すべての顧客のビジネスを成功に導くために、信頼できるサービスをタイムリーに提供するように Micro Focus はワールドワイドのサポート体制を整えています。

保守およびサポート契約を結んだすべてのお客様、および製品を評価中のお客様は、カスタマ サポートを受けることができます。弊社の熟練したスタッフが、可能な限り迅速に専門家としてお客様の質問にお答えします。

<http://supportline.microfocus.com/assistedservices.asp> にアクセスするか、またはメールを supportline@microfocus.com に送信して、Micro Focus SupportLine と直接連絡できます。

また、<http://supportline.microfocus.com> の Micro Focus SupportLine では、最新のサポートに関するニュースや、さまざまなサポート情報を得ることができます。このサイトに初めてアクセスした場合は、ユーザー登録が必要な場合があります。

## Micro Focus SupportLine で必要な情報

Micro Focus SupportLine に連絡する際、可能な場合は以下の情報も提供してください。詳細な情報をご提供いただければ、Micro Focus SupportLine はより効果的なサポートが可能になります。

- 問題の原因と考えられるすべての製品の名前とバージョン番号。
- コンピュータのメーカーと機種。
- オペレーティング システム名とバージョン、プロセッサ、メモリの詳細などのシステム情報。
- 問題を再現する手順などの、問題の詳細な説明。
- 関係するエラー メッセージの正確な表現。
- シリアル番号。

これらの番号を見つけるには、Micro Focus から受信した Electronic Product Delivery Notice 電子メールの件名および本文を参照してください。

# 製品通知サービス

製品通知サービスはシステムトレイで実行されるアプリケーションで、SilkTestの更新が入手可能になった場合にそれを知らせます。また、更新ページに移動可能なリンクも提供します。

## サービスの実行

システムトレイで、更新通知アイコンをクリックすると、製品通知サービスアプリケーションが起動します。

**インストール済みバージョン** 現在インストールされている SilkTest アプリケーションのバージョン番号を提供します。

**更新バージョン** 利用可能な場合、次回のマイナー更新のリンクとバージョン番号を提供します。

**新しいバージョン** 利用可能な場合、次回のフルリリースのリンクとバージョン番号を提供します。

**設定** **設定** ボタンをクリックして、**設定** ウィンドウを開きます。通知サービスで更新をチェックするかどうかと、その頻度を選択します。

# 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。

# SilkTest 製品スイート

SilkTest 製品スイートには、以下のコンポーネントが含まれています。

- SilkTest Workbench : SilkTest Workbench は、新しいネイティブ品質テスト環境です。上級者用の .NET スクリプトと、テストのアクセス可能性を高める革新的なストーリーボードベースのビジュアルテストが提供されます。
- Silk4NET : Silk4NET Visual Studio プラグインを使用すると、Visual Studio で直接 Visual Basic または C# のテスト スクリプトを作成できます。
- Silk4J : Silk4J Eclipse プラグインを使用すると、Eclipse 環境で直接 Java ベースのテスト スクリプトを作成できます。
- SilkTest Recorder : SilkTest Recorder を使用すると、GUI を使用してテストを記録および再生し、それらのテストを SilkTest Classic、Silk4J、または Silk4NET にエクスポートできます。
- SilkTest Classic : SilkTest Classic は、従来の 4Test SilkTest 製品です。
- SilkTest Agent : SilkTest Agent は、テストのコマンドを GUI 固有のコマンドに変換するソフトウェアプロセスです。つまり、テストするアプリケーションをエージェントが動かす、監視しています。ホストマシン上で 1 つのエージェントをローカルに実行できます。ネットワーク環境では、任意の数のエージェントをリモートマシン上で実行できます。

インストールする製品スイートによって、使用できるコンポーネントが決まります。すべてのコンポーネントをインストールするには、完全インストール オプションを選択します。SilkTest Classic を除くすべてのコンポーネントをインストールするには、標準インストール オプションを選択します。

# 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 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。

サンプル アプリケーションを <http://supportline.microfocus.com/websync/SilkTest.aspx> からダウンロードします。サンプル アプリケーションをインストールしたら、**スタート > プログラム > Silk > SilkTest > Samples > Silk4NET Samples** をクリックして、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。

<b>Insurance Co. Web サイト</b>	<a href="http://demo.borland.com/InsuranceWebExtJS/">http://demo.borland.com/InsuranceWebExtJS/</a>
<b>Green Mountain Outpost Web</b>	<a href="http://demo.borland.com/gmopost/">http://demo.borland.com/gmopost/</a>

# 動的オブジェクト解決

動的オブジェクト解決により、オブジェクトを検索し識別する XPath クエリを使用した、テスト メソッドの記述が可能になります。動的オブジェクト解決は、テスト メソッド内でオブジェクトを識別するために、Find メソッド、または FindAll メソッドを使用します。たとえば、以下のクエリは、特定のウィンドウの子である、キャプションが「ok」である最初のボタンを見つけます。

```
Dim okButton = window.find("//PushButton[@caption=ok]")
```

動的オブジェクト解決がうまく機能するテスト環境の種類は以下のとおりです。

- GUI が変更中であるアプリケーション環境

たとえば、メニューやダイアログ名が変更される可能性があるメニューに属するダイアログの **Check Me** チェック ボックスをテストする場合、動的オブジェクト解決を使用すると、メニューやダイアログ名に依存せずにチェック ボックスをテストすることができます。この場合、正しいコンポーネントをテストしたことを保障するために、チェック ボックス名、ダイアログ名、メニュー名を検証することができます。

- 動的なテーブルやテキストを含んだ Web アプリケーション

たとえば、Web ページ上のある項目をユーザーがポイントするときだけ表示されるテーブルをテストするために動的オブジェクト解決を使用すると、テーブルを表示するためにページのどの部分をクリックする必要があるかに関係なく、テーブルを見つけて利用するテスト メソッドを作成できます。

- ビューを使用する Eclipse 環境

たとえば、ビュー コンポーネントを使用する Eclipse 環境をテストするために動的オブジェクト解決を使用すると、そのビューより前に開く必要のあるオブジェクトの階層に関係なく、ビューを識別することができます。

## 動的オブジェクト解決を利用する利点

動的オブジェクト解決を利用する利点は次の通りです：

- 動的オブジェクト解決は、W3C（World Wide Web Consortium）によって定義された共通の XML ベース言語である XPath クエリ言語のサブセットを使用します。
- 動的オブジェクト解決は、テストしているアプリケーションのオブジェクトのリポジトリというよりは、単一のオブジェクトを必要とします。XPath クエリを使用すると、テストケースにおいて、Find コマンド（サポートされている XPath 構成子を続ける）を使用して、オブジェクトを見つけることができます。

## XPath の基本概念

Silk4NET は、XPath クエリ言語のサブセットをサポートしています。XPath に関する追加の情報については、<http://www.w3.org/TR/xpath20/> を参照してください。

### 基本概念

XPath 式は現在のコンテキスト、つまり、Find メソッドを呼び出したオブジェクトの階層上における位置に依存します。ファイル システムと同じように、すべての XPath 式は、この位置に依存します。例：


- "//Shell" は、現在のオブジェクトに相対的なすべての階層にあるすべての Shell を見つけます。
- "Shell" は、現在のオブジェクトの直下の子であるすべての Shell を見つけます。

さらに、ある XPath 式は、コンテキストの影響を受けます。たとえば、myWindow.find(xpath) は、myWindow が現在のコンテキストとなります。

## サポートする XPath のサブセット

Silk4NET は、XPath クエリ言語のサブセットをサポートしています。FindAll または Find コマンド (サポートする構成子を続ける) を使用して、テストケースを作成します。

以下の表には、Silk4NET がサポートする構成子が一覧されています。

サポートする XPath 構成子	サンプル	説明
属性	MenuItem[@caption='abc']	現在のコンテキストの子のオブジェクト定義内で、指定した caption 属性を持つすべての MenuItem を見つけます。次の属性がサポートされています: caption (caption index なし)、priorlabel (index なし)、windowid。
索引	MenuItem[1]	現在のコンテキストの子のうち、最初の MenuItem を見つけます。XPath 内での索引は 1 から始まります。
論理演算子: and、or、not、=、!=	MenuItem[not(@caption='a' or @windowid!='b') and @priorlabel='p']	
の項目です。	TestApplication.Find("//Dialog[@caption='Check Box']/../.")	Find コマンドが実行されるコンテキストを見つけてます。たとえば、このサンプルは、 TestApplication.Find("//Dialog[@caption='Check Box']") と同じ意味になります。
..	Desktop.Find("//PushButton[@caption='Previous']/../PushButton[@caption='Ok']")	オブジェクトの親を見つけてます。たとえば、このサンプルは、caption が "Previous" である PushButton と同列にある caption が "Ok" である PushButton を見つけます。
/	/Shell	現在のオブジェクトの直下の子であるすべての Shell を見つけます。  <b>注:</b> 「/Shell」は、「Shell」と同じです。
/	/Shell/MenuItem	現在のオブジェクトの子であるすべての MenuItem を見つけます。
//	//Shell	現在のオブジェクトに相対的なすべての階層にあるすべての Shell を見つけます。
//	//Shell//MenuItem	現在のオブジェクトの直下の子である Shell の直接あるいは間接的な子であるすべての MenuItem を見つけます。
//	//MenuItem	現在のコンテキストの直接あるいは間接的な子であるすべての MenuItem を見つけます。

サポートする XPath 構成子	サンプル	説明
*	*[@caption='c']	現在のコンテキストの直下の子であり、かつ指定した caption を持つすべてのオブジェクトを見つけます。
*	//MenuItem/*/Shell	MenuItem の孫であるすべての Shell を見つけます。

以下の表には、Silk4NET がサポートしていない XPath 構成子が一覧されています。

サポートしない XPath 構成子	例
右辺、左辺ともに属性を指定して比較する。	PushButton[@caption = @windowid]
属性名を右辺に指定することはサポートされません。属性名は左辺に指定する必要があります。	PushButton['abc' = @caption]
複数の XPath 式を 'and' あるいは 'or' で結合する。	PushButton [@caption = 'abc'] or .//Checkbox
複数の属性をかぎ括弧で指定する。	PushButton[@caption = 'abc' [@windowid = '123'] (代わりに PushButton [@caption = 'abc and @windowid = '123'] を使用)
複数の索引をかぎ括弧で指定する。	PushButton[1][2]
クラスあるいはクラス名の一部にワイルドカードを含むクラス ワイルドカードを明示的に指定しない構成子。	//[@caption = 'abc'] (代わりに //*[@caption = 'abc'] を使用) "//*Button[@caption='abc']"

## XPath のサンプル


以下の表示では、サンプルの XPath クエリの一覧を提供し、それぞれのクエリの意味を説明します。

XPath 文字列	説明
desktop.find("/Shell[@caption='SWT Test Application']")	指定した caption を持つ最初のトップレベルの Shell を見つけます。
desktop.find("//MenuItem[@caption='Control']")	指定した caption を持つ任意の階層における MenuItem を見つけます。
myShell.find("//MenuItem[@caption!='Control']")	指定した caption を持たない myShell の任意の子階層における MenuItem を見つけます。
myShell.find("Menu[@caption='Control']/MenuItem[@caption!='Control']")	指定した Menu を親とし、さらに myShell をその親とする、指定した MenuItem を見つけます。
myShell.find("//MenuItem[@caption='Control' and @windowid='20']")	指定した caption と windowid を持つ myWindow の任意の子階層における MenuItem を見つけます。
myShell.find("//MenuItem[@caption='Control' or @windowid='20']")	指定した caption または windowid を持つ myWindow の任意の子階層における MenuItem を見つけます。

XPath 文字列	説明
<code>desktop.findAll("/Shell[2]/*/PushButton")</code>	親として 2 番目のトップレベルの Shell を持つ任意の親を持つすべての PushButton を見つけます。
<code>desktop.findAll("/Shell[2]//PushButton")</code>	直接あるいは間接的な親として 2 番目の Shell を使用するすべての PushButton を見つけます。
<code>myBrowser.find("//FlexApplication[1]//FlexButton[@caption='ok']")</code>	指定したブラウザ内の最初の FlexApplication 内にある最初の FlexButton を見つけます。
<code>myBrowser.findAll("//td[@class='abc*']//a[@class='xyz']")</code>	値 abc* の属性 class を持つ td 要素の直接あるいは間接的な子である、値 xyz の属性 class を持つすべてのリンク要素を見つけます。

## XPath のパフォーマンス問題のトラブルシューティング

複雑なオブジェクト構造を持つアプリケーションをテストする場合、パフォーマンスの問題やスクリプトの信頼性に関する問題が発生する場合があります。このトピックでは、記録中に Silk4NET が自動的に生成したロケータとは異なるロケータを使用することによって、スクリプトのパフォーマンスを改善させる方法について説明します。

 **注:** 一般に、複雑なロケータを使用することは推奨しません。複雑なロケータを使用すると、テストの信頼性を損なう可能性があります。複雑なロケータは、テストアプリケーションの構造をほんの少し変更しただけで機能なくなってしまう可能性があります。それにもかかわらず、スクリプトのパフォーマンスが要求を満たしていない場合には、より固有のロケータを使用することによってテストのパフォーマンスを向上できる可能性があります。

例として、MyApplication アプリケーションの要素ツリーを以下に示します。

```
Root
  Node id=1
    Leaf id=2
    Leaf id=3
    Leaf id=4
    Leaf id=5
  Node id=6
    Node id=7
      Leaf id=8
      Leaf id=9
    Node id=9
      Leaf id=10
```

以下の最適化手法のいくつかを使用して、スクリプトのパフォーマンスを改善させることができます。

- 複雑なオブジェクト構造内の要素を特定したい場合は、オブジェクト構造全体ではなく、その特定の部分だけを検索するようにします。たとえば、サンプルツリーの識別子 4 を持つ要素を検索する場合に `Root.Find("//Leaf[@id='4']")` というクエリーを使用している場合、`Root.Find("/Node[@id='1']/Leaf[@id='4']")` というクエリーで置き換えます。最初のクエリーでは、識別子 4 を持つリーフが、アプリケーションの要素ツリー全体から検索されます。最初のリーフが見つかった時点で返されます。2 番目のクエリーでは、識別子 1 を持つノードと識別子 6 を持つノードがある最初のレベルのノードがまず検索された後、識別子 4 を持つすべてのリーフが識別子 1 を持つノードのサブツリー内から検索されます。
- 同じ階層内の複数の項目を特定したい場合は、まずは階層を特定してからループ内で項目を特定します。`Root.FindAll("/Node[@id='1']/Leaf")` というクエリーを使用している場合、次のようなループで置き換えます。

```
Public Sub Main()
  Dim node As TestObject
```

```
node = _desktop.Find("//Node[@id='1']")
For i As Integer = 1 To 4 Step 1
    node.Find("/Leaf[@id='"+i+"']")
Next
End Sub
```

# xBrowser のよくある質問

このセクションでは、Web アプリケーションをテストするときに発生することがある質問のコレクションを示します。

## 要素のテキストに使用されるフォント タイプの確認方法

属性名を「:」で区切ると、DOM 要素の `currentStyle` 属性のすべての属性にアクセスできます。

**Windows Internet Explorer 8 以前**      `wDomElement.GetProperty("currentStyle:fontName")`

**Windows Internet Explorer 9 またはそれ以降および Mozilla Firefox などの他のすべてのブラウザ**      `wDomElement.GetProperty("currentStyle:font-name")`

## textContents、innerText、および innerHtml の違い

- `textContents` は、書式設定のみを目的とする要素およびその子要素に含まれるすべてのテキストです。
- `innerText` は、要素およびその子要素に含まれるすべてのテキストを返します。
- `innerHTML` は、要素に含まれるすべてのテキスト (html タグも含む) を返します。

以下の html コードについて検討します。

```
<div id="mylinks">
  This is my <b>link collection</b>:
  <ul>
    <li><a href="www.borland.com">Bye bye <b>Borland</b> </a></li>
    <li><a href="www.microfocus.com">Welcome to <b>Micro Focus</b> </a></li>
  </ul>
</div>
```

以下の表に、返されるプロパティの詳細を示します。

コード	返される値
<code>browser.DomElement("//div[@id='mylinks']").GetProperty("textContents")</code>	This is my link collection:
<code>browser.DomElement("//div[@id='mylinks']").GetProperty("innerText")</code>	This is my link collection:Bye bye Borland Welcome to Micro Focus
<code>browser.DomElement("//div[@id='mylinks']").GetProperty("innerHTML")</code>	This is my <b>link collection</b>: <ul> <li><a href="www.borland.com">Bye bye <b>Borland</b></a></li> <li><a href="www.microfocus.com">Welcome to <b>Micro Focus</b></a></li> </ul>



## innerText をカスタム クラス属性として構成したが、ロケータで使用されない

ロケータ文字列に使用する属性には最大長があります。InnerText は長くなりすぎる傾向があり、ロケータで使用できない場合があります。可能な場合は、textContent を代わりに使用してください。

## クロスブラウザ スクリプトの作成時に必要な処置

クロスブラウザ スクリプトを作成する場合は、以下の 1 つまたは複数の問題に遭遇する場合があります。

- 属性値が異なる。たとえば、Windows Internet Explorer の色が "# FF0000" として、Mozilla Firefox の色が "rgb(255,0,0)" として返されます。
- 属性名が異なる。たとえば、Windows Internet Explorer 8 以前のバージョンではフォント サイズ属性が "fontSize" と呼ばれ、Windows Internet Explorer 9 以降および Mozilla Firefox などの他のすべてのブラウザでは "font-size" と呼ばれます。
- 一部のフレームワークで異なる DOM ツリーがレンダリングされることがある

## スクリプトで Firefox と Internet Explorer を区別する方法

- BrowserApplication クラスには、「Firefox」または「Internet Explorer」を返すプロパティ "browserType" があります。これをロケータに追加すると、いずれのブラウザに一致させるかを定義できます。
- BrowserWindow には、現在のウィンドウのユーザー エージェント文字列を返すメソッド GetUserAgent があります。

## 安定したクロスブラウザ テストを実現するために最適なロケータ

組み込みロケータ生成プログラムでは、安定したロケータの作成が試みられます。ただし、情報を使用できない場合、高品質のロケータを生成することは困難です。この場合、ロケータ生成プログラムでは、階層形式の情報およびインデックスが使用されます。その結果、直接的な記録/再生には適していても、安定した日常的な実行には適さない脆弱なロケータが生成されます。さらに、クロスブラウザ テストでは、いくつかの AJAX フレームワークで異なるブラウザに対して異なる DOM 階層がレンダリングされることがあります。

この問題を回避するには、アプリケーションの UI 要素にカスタム ID を使用します。

## アプリケーションのログ出力に正しくないタイムスタンプが含まれる

この方法によって、同期に関して予期しない結果が発生する場合があります。この問題を回避するには、HTML 同期モードを指定します。

## 新しいページに移動したあと、スクリプトがハングする

この問題は、AJAX アプリケーションによりブラウザがビジー（サーバー プッシュ/ActiveX コンポーネントの接続が開いている）のままになっている場合に、発生することがあります。HTML 同期モードを設定してください。他のトラブルシューティングのヒントについては、「xBrowser のページ同期」のトピックを参照してください。

## 正しくないロケーターが記録されている

マウスを要素上に移動したときに、要素の属性が変更することがあります。Silk4NET によってこのシナリオの追跡が試行されますが、失敗することがあります。影響を受ける属性を特定し、それが Silk4NET で無視されるように構成してください。

## Windows Internet Explorer で要素を囲む四角形の位置が正しくない

- 拡大率が 100% に設定されていることを確認します。このようにしないと、四角形が正しく配置されません。
- ブラウザ ウィンドウの上に通知バーが表示されていないことを確認します。Silk4NET では、通知バーを処理できません。

## Link.Select で、Internet Explorer で新しく開いたウィンドウにフォーカスが設定されない

この制限は、ブラウザの構成設定を変更することで修正できます。新しく開いたウィンドウが常にアクティブ化されるようにオプションを設定します。

## DomClick(x, y) が Click(x, y) のように動作しない

アプリケーションで onclick イベントを使用しており、座標を必要とする場合、DomClick メソッドは動作しません。代わりに、Click を使用します。

## FileInputField.DomClick() でダイアログが開かない

代わりに、Click を使用します。

## マウス移動設定がオンになっているにもかかわらず、すべての操作が記録されない理由

多くの無用な MoveMouse 操作がスクリプトに影響を及ぼさないように、Silk4NET では以下の操作が行われます。

- マウスが一定時間静止している場合にのみ、MoveMouse 操作が記録されます。

- マウスを要素上に移動したあとで操作が行われていることが確認された場合にのみ、MoveMouse 操作が記録されます。場合によっては、スクリプトに手動操作を追加することが必要となることがあります。

## xBrowser API で公開されていない機能が必要な場合の対処方法

ExecuteJavaScript() を使用して、JavaScript コードを Web アプリケーションから直接実行できます。この方法は、ほとんどすべての問題の回避策となります。

## ロケーターでクラスとスタイルの属性が使用されない理由

これらの属性は AJAX アプリケーションで頻繁に変更され、ロケーターの安定性が損なわれることがあります。そのため、無視リストに含まれています。ただし、多くの場合、これらの属性を使用してオブジェクトを識別できるため、アプリケーションで使用することに意味がある場合があります。

## 再生中にダイアログが認識されない

スクリプトを記録するときに、Silk4NET はいくつかのウィンドウを Dialog として認識します。スクリプトをクロス ブラウザ スクリプトとして使用する場合は、ブラウザによっては Dialog が認識されないため、Dialog を Window に置き換える必要があります。

たとえば、スクリプトに以下の行があるとします。

```
/BrowserApplication//Dialog//PushButton[@caption='OK']
```

クロス ブラウザ テストを可能にするには、次のように行を書き換えます。

```
/BrowserApplication//Window//PushButton[@caption='OK']
```

# 索引

## A

AJAX アプリケーション  
スクリプトのハング 26

## C

Chrome  
クロスブラウザ スクリプト 25  
Customer Care 4

## F

FAQ  
xBrowser 24  
Firefox  
クロスブラウザ スクリプト 25  
Firefox  
IE との区別 25  
ロケーター 25

## I

innerHTML 24  
innerText 24, 25  
Internet Explorer  
Firefox との区別 25  
link.select のフォーカスの問題 26  
ロケーター 25  
Internet Explorer  
クロスブラウザ スクリプト 25

## S

SupportLine 4

## T

textContent 24

## W

WebSync 4  
Works Order 番号 4

## X

xBrowser  
DomClick が Click のように動作しない 26  
FAQ 24  
FieldInputField.DomClick でダイアログが開かない  
26  
innerText がロケーターで使用されない 25  
link.select のフォーカスの問題 26  
textContent、innerText、innerHTML 24

新しいページへの移動 26  
タイムスタンプ 25  
正しくないロケーターの記録 26  
フォント タイプの検証 24  
ブラウザの種類の違い 25  
ロケーターにないクラスとスタイル 27  
ロケーターの記録 25

## xBrowser

Windows Internet Explorer で四角形の位置が正しく  
ない 26  
機能の公開 27  
クロスブラウザ スクリプト 25  
認識されないダイアログ 27  
マウス移動の記録 26

## XPath

概要 19  
基本概念 19  
サポートされているサブセット 20  
サンプル 21  
トラブルシューティング 22  
XPath のトラブルシューティング 22

## か

カスタマー ケア 4

## こ

更新 5

## さ

再生  
認識されないダイアログ 27

## し

シリアル番号 4

## せ

製品サポート 4  
製品の更新 5

## た

ダイアログ  
認識しない 27  
タイムスタンプ 25  
ダウンロード 4

## て

テスト ケース  
サンプル クエリ 21

## と

動的オブジェクト解決  
概要 19  
サンプル クエリ 21

## れ

連絡先情報 4

## ろ

ロケーター

xBrowser 25  
xBrowser 内で不正 26

## わ

ワーク オーダー番号 4